

MAGAZINE MSX
AÑO I
Núm. 8
Diciembre
1985
300 Ptas.
TEST:
DYNADATA
DPC-200
Sonido
Código máquina
Libros, trucos...
Estudiamos
la memoria
de video
CD-ROM:
BASE DE DATOS
EN CASA



Editorial

A partir de este mes, MSX Magazine será más cara. La subida se debe a una serie de factores que no son ajenos a la revista, el papel es más caro, etc. Por esta razón, nosotros nos vemos en el compromiso de intentar mejorar el producto final. Para ello, este mes iniciamos dos secciones nuevas que esperamos sean del agrado de los lectores; Test y libros.

En la primera sección comentaremos las innovaciones referentes a ordenadores y periféricos que aparezcan en el mercado, mientras que en la segunda sección la dedicaremos exclusivamente a la crítica de libros. Insistimos en la necesidad de que seaís vosotros los que con vuestras críticas nos ayudeis a mejorar la revista.

Y aprovechando la ocasión, queremos felicitar a CITEMA, por el 25 aniversario de SIMO. La feria ha mostrado, una vez más, la importancia de los ordenadores personales y los ordenadores domésticos en la vida cotidiana.

En lo que a MSX se refiere, cabe destacar la aparición ¡por fin! del Dragón MSX y la creación, por parte de varias casas de ordenadores, de aulas informáticas. Efectivamente, Eurohard, S.A. ha presentado una maqueta del tan oído y comentado MSX de Dragón, uniéndose así a la corriente actual de la estandarización. El citado aparato saldrá a la venta a partir de finales de enero a un precio de unas 55.000 Ptas. Sus características principales son similares al resto de los ordenadores de la misma norma.

En cuanto a las aulas informáticas, tanto Toshiba como Dynadata, han presentado las suyas, basadas en un ordenador principal y 32 puestos adicionales. Cada puesto, dispondrá de un teclado y su correspondiente pantalla, con el profesor dirigiendo los estudios desde su ordenador maestro (que será un MSX). A partir de este momento, la lucha se llevará a cabo en las aulas, ya que con la aparición de estas dos marcas, junto con la que se rumorea de Philips, hay que unir la que ya posee Spectravideo, que tiene a su favor el haber preparado y lanzado su aula informática con bastante antelación.

No podemos dejar pasar estas fechas sin enviar a la gran familia de usuarios de MSX un sincero saludo, deseando que la Navidad-85 llegue cargada de felicidad para todos.

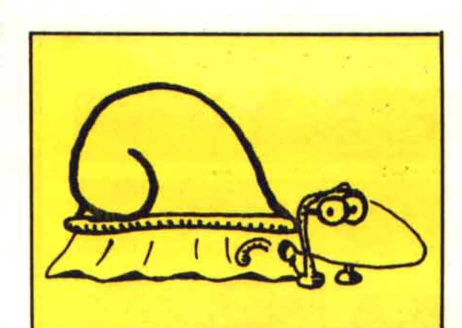
Desde aquí, todos los que hacemos posible que esta revista llegue puntualmente a tu kiosko cada mes, deseamos a nuestros lectores y anunciantes unas felices fiestas y un próspero —a pesar del IVA— año 1986.

s.i.m.o.
25 años



SUMARIO







LIBROS

NOTICIAS

Dro Soft y sus Programas para MSX

En breve plazo veremos unos buenos programas que ya se comercializan para otros ordenadores y que son muy entretenidos. Se trata de ARCHON y M.U.L.E. el primero es una especie de ajedrez, con piezas que habrá que mover y enfrentar a las del enemigo. Obviamente, el fin del juego es el de atrapar, en este caso, el brujo o la bruja contraria. Sin embargo, en M.U.L.E., un juego para cuatro personas, tendrá que hacerse cargo de una parcela en el planeta X y mediante unas máquinas especialmente diseñadas por cada jugador, sobrevivir a todo... Con estos programas, Dro Soft inicia su etapa en el siempre inconforme mercado MSX.

Nueva red local de Dynadata

La Red MSX de Dynadata S.A., para aulas informáticas de enseñanza se basa en una unidad controladora que contiene de una a cuatro unidades de diskettes de 360 ó 720 Kbytes cada una, capaz de soportar y dirigir 33 puestos de trabajo simultáneos, para la transmisión de datos y programas entre el profesor y el alumno. Cada puesto, tanto el del profesor como el del alumno, está compuesto por un ordenador MSX, el cual está conectado al controlador por medio de una tarjeta de interface. Igualmente, cada puesto dispone de un monitor en fósforo verde o en color y el del alumno de una unidad de cassette. Los alumnos y el profesor comparten las unidades de diskette del controlador y una impresora de 180 c.p.s. conectada a éste. Además, simultáneamente al lanzamiento de la red, se comecializará un completo curso de BASIC y todas las asignaturas desde primero hasta octavo de E.G.B. en soporte informático, incluyendo importantes lenguajes de programación como LOGO, FORTRAN, PASCAL y COBOL.



Philips potencia el mercado

Que Philips se está volcando hacia el mercado de MSX no es nada nuevo, pero sí merece la pena destacar las importantes innovaciones en cuanto al software se refiere. La comercialización de un procesador de textos y una Base de Datos, para ser utilizado por un ordenador con unidad de diskettes (aunque también viene con un cassette) viene a engrosar el mercado de software para este estándar, aunque hay que destacar que es único en su línea para aplicaciones.

Los dos programas están pensados para facilitar considerablemente el trabajo a quienes los utilizan, además hay que destacar que ambos se interrelacionan, es decir, con la Base de Datos podemos almacenar direcciones, mientras que con el tratamiento de textos podemos escribir cartas utilizando las direcciones de la Base de Datos (en números futuros comentaremos los dos programas a fondo ya que merecen un punto y aparte). También se comercializará un lenguaje muy interesante: el LOGO. Este viene en cartucho, con un manual de 150 páginas que simplifica el proceso de apredizaje.

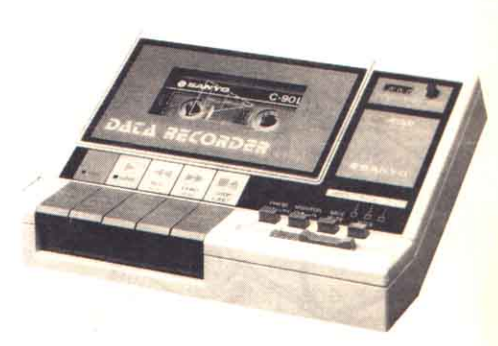
Lo que sí será un logro en adelantos para los ordenadores MSX va a ser, sin lugar a dudas, la red local que a principios de año van a comercialzar esta firma. La posibilidad de poder conectar 32 ordenadores MSX no es nada nuevo, al fin y al cabo, hace varios meses ya se presentó la primera red local para estos ordenadores, sin embargo, poder tener una red con PC de IBM y ordenadores MSX, es algo que revolucionará el mercado, desde las aulas informáticas hasta las pequeñas y medianas empresas.

Data cassette de Sanyo

El nuevo periférico, denominado DR 202A, es un cassette nada corriente. Entre sus características podemos destacar un dispositivo de ADSS ó Sistema de Búsqueda de Datos Automático de 3 programas que permite acceder automáticamente a un programa, debido principalmente a que cuando se graba con este aparato, automáticamente se deja un espacio entre ellos. Además posee un interruptor de cambio de fase, lo que permite la utilización de cualquier tipo de cinta comercial.

Los controles, situados en el panel frontal, permiten un manejo fácil, sencillo y rápido.

El nuevo aparato con un precio de 11.000 Ptas., lo podemos situar, gracias a la posibilidad de búsqueda de programas, a medio camino entre el diskette y la cinta magnética.

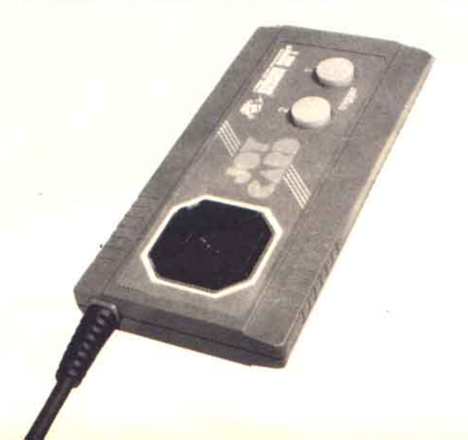


Nuevas impresoras de Citizen

La archiconocida empresa relojera Citizen, ya no sólo se dedica a fabricar relojes, también se dedica a hacer impresoras.

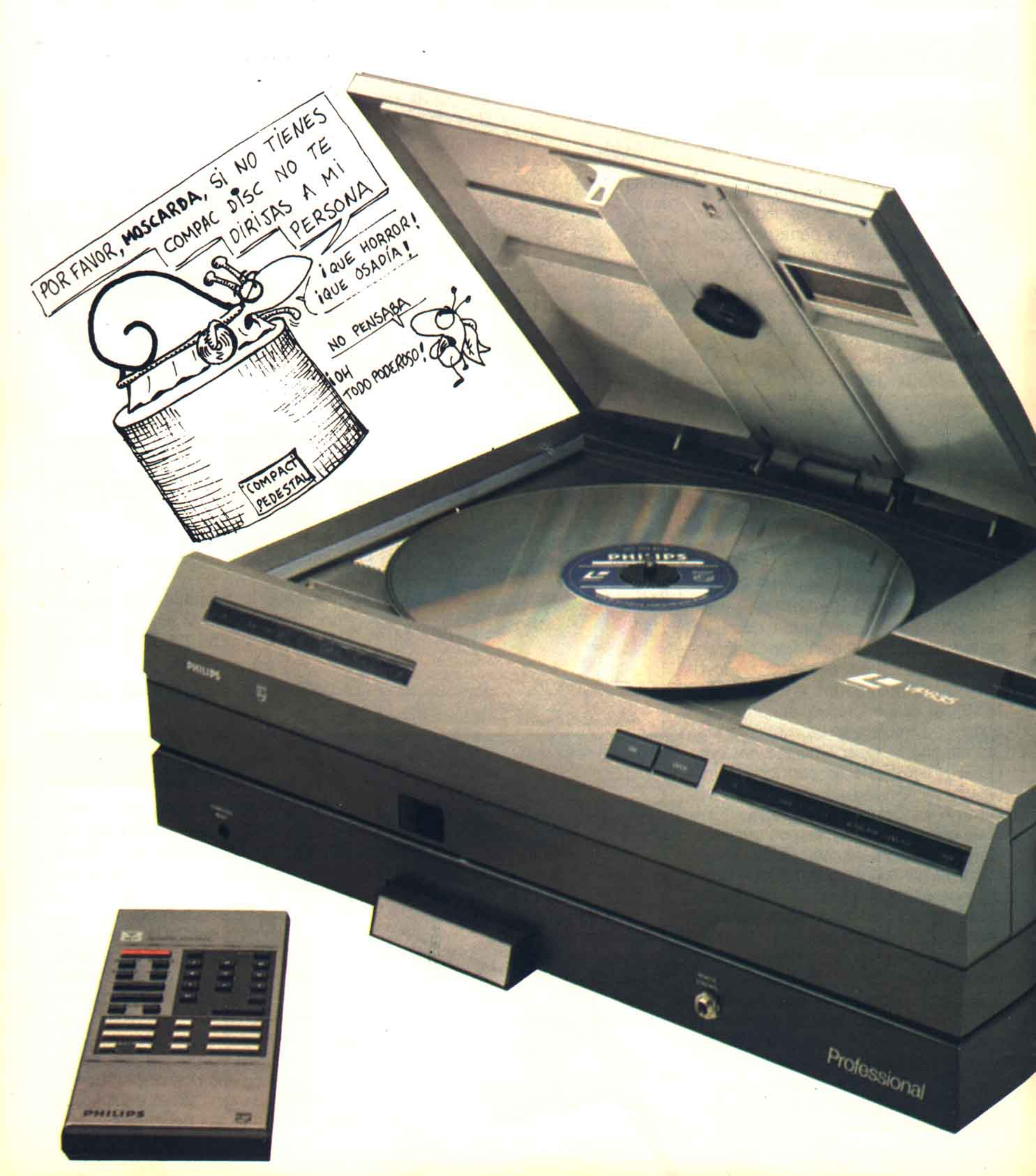
Efectivamente, Citizen Europe, Ltd., la nueva filial de Citizen Watch Company de Japón, ha introducido recientemente una nueva gama de impresoras con matriz por puntos, conocidas como MSP-10, MSP-15, MSP-20, MSP-25, 120-D y Premier 35.

Joycard: nuevo concepto en joysticks



Fabricado por Hudson Soft y distribuido por Serma para todo tipo de ordenadores domésticos, este nuevo joystick destaca por lo completo que es. A un lado, nos encontramos con dos disparadores siempre necesarios para juegos del tipo Hyper Sport de Kanami, mientras que en la parte opuesta tenemos las cuatro teclas del cursor, ideal para juegos que requieren una acción combinada de las teclas. Los dos teclados se complementan, así en unos casos necesitaremos sólo los dos disparadores, mientras que en otros echaremos mano de las teclas del cursor.

Compac Dis



POR FAVOR, MOSCARDA, SI NO TIENES COMPAC DISC NO TE DIRIJAS A MI PERSONA
¡QUE HORROR! ¡QUE OSADÍA!
NO PENSABA
¡OH TODO PODEROSO!
COMPACT PEDESTAL
PHILIPS
PHILIPS
Professional

C: nuevo concepto en memoria

Con la llegada en un futuro no muy lejano de la 2.ª Generación de MSX, aparecerá también un periférico que revolucinará el tratamiento, almacenamiento y grabación de datos: el Compact Disc.

Ae todos los artilugios digitales de grabación de datos desarrollados hasta la fecha, el *Compact Disc* es el más increíble. Resulta igualmente increíble el hecho de que haya tardado nada menos que cuatro años, desde que la tec-



nología del *Compact Disc* apareció en el mercado, hasta su aplicación en los ordenadores. A finales de 1984, la industria nipona empezó a especular con las posibilidades de almacenamiento de programas y datos en este nuevo soporte. El objeto de tal especulación resultó ser la aparición de los *CD-ROM (ROM* en *Compact Disc).* Brevemente diremos que el *CD-ROM* posee las ventajas del almacenamiento masivo en disco con un rápido acceso a ellos de la memoria *ROM*, es decir se pueden almacenar 540 *Mbytes* de información (un *Mbyte* equivale a 1,048,576 *bytes*), con un acceso relativamente rápido.

Tecnología digital

La grabación en *Compact Disc* es un concepto totalmente nuevo en el ámbito del almacenamiento, reproducción y grabación de datos. Los discos convencinales, ya sean los antiguos cilindros o los modernos discos de vinilo, almacenan el sonido original como su equivalente valor análogo que modula la pared del surco. Una aguja que puede ser metálica, de zafiro o diamante (según la edad del tocadiscos), recoge información del surco modulado y la transfiere al aire (mediante altavoces).

El Compact Disc realiza este proceso de forma digital. El sonido original se convierte en una corriente de palabras digitales que se corresponden con su voltaje instantáneo o nivel de señal de audio. A continuación, ésta corriente de palabras se almacena en un disco, en pistas espirales que comprimen las irregularidades de la superficie del disco.

Combinando las irregularidades no reflectantes con las áreas del disco totalmente lisas, un sistema de lectura óptica (mediante láser), podrá leer el disco y obtener información. A continuación, un sistema basado en el microprocesador del ordenador, reconvierte la señal digital en señal de sonido. Un *Compact Disc* es capaz de grabar, en una cara de un disco de menos de 5 pulgadas de diámetro, toda la información necesaria para almacenar y reproducir el equivalente de una hora de música a un nivel muy superior al obtenido de un disco LP convencional. Para obtener una fiabilidad de la reproducción de un *Compact Disc*, la señal analógica de audio ha de ser analizada (para convertirla en su equivalente digital) a una velocidad superior al doble de la frecuencia más alta de reproducir. Los sistemas de audio de alta fidelidad requieren un límite de frecuencia máxima del orden de los 15KHz, de forma que la frecuencia analizadora ha de ser, al menos de 30KHz. Una hora de reproducción de música requerirá por lo menos 60 (minutos) × 60 (segundos) × 30.000 (análisis por segundo), ó 108 millones de análisis digitalizados. Cada análisis requiere dos *bytes* de datos en el disco para una resolución adecuada y las técnicas de correciones de errores requieren una cantidad superior de datos.

La increíble capacidad de almacenamiento de datos en *Compact Disc* se debe a la finura de las pistas (menos de una micra de ancho). Las irregularidades de las pistas tienen el mismo tamaño que la longitud de

onda del rayo láser utilizado para leer la información.

Aunque los *Compact Disc* almacenan los dastos de forma que se pueden convertir en señales de audio, esta misma tecnología se puede utilizar para almacenar datos necesarios para el ordenador. Los trabajos realizados en *CD-ROM* se deben a casas importantes, como *Philips, Sony* y *Pioneer*, desarrollando una estándar en este nuevo periférico. Los detalles del interface de *Philips* y *Sony* difieren en algunos aspectos (mientras que de *Pioneer* no teníamos suficiente información), pero el formato de datos y capacidad de almacenamiento son los mismo: 540 *Mbytes* de datos por disco.

Almacenamiento masivo de datos

Aquí entramos en el terreno de la especulación. Los mecanismos necesarios para la reproducción de los *CD-ROM* son los mismos que los *Compact Disc*, excepto que el circulo A/D (Analógico/Digital), no es necesario y el precio de este periférico deberá rondar el de la unidad de discos. El coste de los discos *CD-ROM* dependerá del mercado potencial, estando a la altura de los discos convencionales.

Los 540 *Mbytes* de un sólo disco, permitirá a los fabricantes de ordenadores juntar todo el *software* de aplicación necesario para un cliente en un único disco.

Comparado el coste de la grabación con el de otros soportes magnéticos (discos o cintas), el *CD-ROM* sale bastante más rentable. Un sólo disco *CD-ROM* puede almacenar fácilmente, *CP-M 21 ó el* MS-DOS (incluyendo el *BIOS*), además del *WordStar, SuperCalc*, un intérprete *BASIC*, compiladores de *Pascal, C* y *Fortran, software* para comunicaciones y las obras completas de Miguel de Cervantes. El problema radica en que los discos *CD-ROM* son unos soportes de lectura, al igual que los chips *ROM* de silicona. Aunque los *CD-ROM* pueden almacenar gran cantidad de información, todavía necesitaremos los *diskettes* para la mayoría de las apicaciones. La gran cantidad de *sotware* existente en el mercado asume el que los ficheros se pueden escribir en el mismo soporte que almacena los datos; esto es, la instrucción *OPEN* Nombre fichero , normalmente asume que el origen y el destino del alamacenamiento masivo de datos es el mismo periférico.

Un Compact Disc de menos de 5 pulgadas puede almacenar hasta un total de 540 Mbytes.

Modificar este tipo de *software* para que el programa y los datos generados por él puedan alamacenarse en soportes separados, no supone problema alguno, pero como en cualquier modificación hay que proceder con cuidado.

Bases de datos en CD-ROM

Algunos tipos de Bases de Datos necesitan actualizar ficheros con frecuencia, otras no tanto. Si la Base de Datos se utiliza para mantener los registros de los clientes o un control de inventario, las actualizaciones se deberán realizar con todas las transacciones. En este caso, el *software* para tratamiento de datos, necesitará reescribirse y la mayoría de las ventajas de la *CD-ROM* se perderán.

Por otro lado, muchas Bases de Datos no necesitan modificación alguna: una pequeña actualización en *diskette* deberá ser suficiente. La *CD-ROM* resulta ideal en un contorno de este tipo.

Es en este tipo de aplicacines donde los intereses de las casas de *software* y las editoriales convergen. El bajo coste de la *CD-ROM*, unido a un amplio mercado de Bases de Datos estádisticas, deberá hacer pensar a las grandes empresas editoriales.

La información de las Bases de Datos generalmente se paga por minuto, los clientes llaman al teléfono adecuado y conectan sus ordenadores por medio de un *modem*. Estas Bases de Datos normalmente funcio-

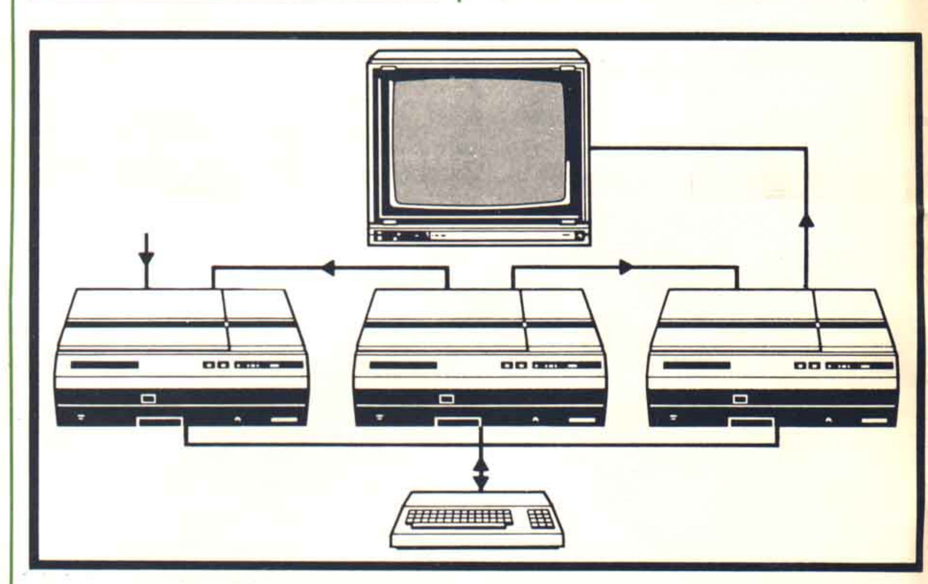


Foto 1. Tres Video Discos con cartuchos EPROM conectados en paralelo al mismo ordenador.

nan con grandes ordenadores y, como tales, los ususarios podrán realizar actualizaciones con frecuencia, sin embargo, cuando no haya que hacer actualización alguna, la *CD-ROM* vendrá «a gusto del consumidor». ¿Porqué pagar el tiempo del ordenador y el teléfono cuando podemos tener la información necesaria utilizando un *Compac Disc* de bajo coste?

Aunque estamos acostumbrados a las Bases de Datos como estructuras dinámicas con alteraciones constantes en los registros y los ficheros, es sorprendente la cantidad de información que permanece estática e inalterable; pensemos en los libros de una bilioteca, sin mencionar la cantidad de datos geográficos, científicos, estadísticos y legales. Las *CD-ROM* harán posibles un nuevo tipo de publicaciones y será interesante ver como los editores tradicionales y casas de software cooperan para optimizar su uso o rechazar esta interesante opción.

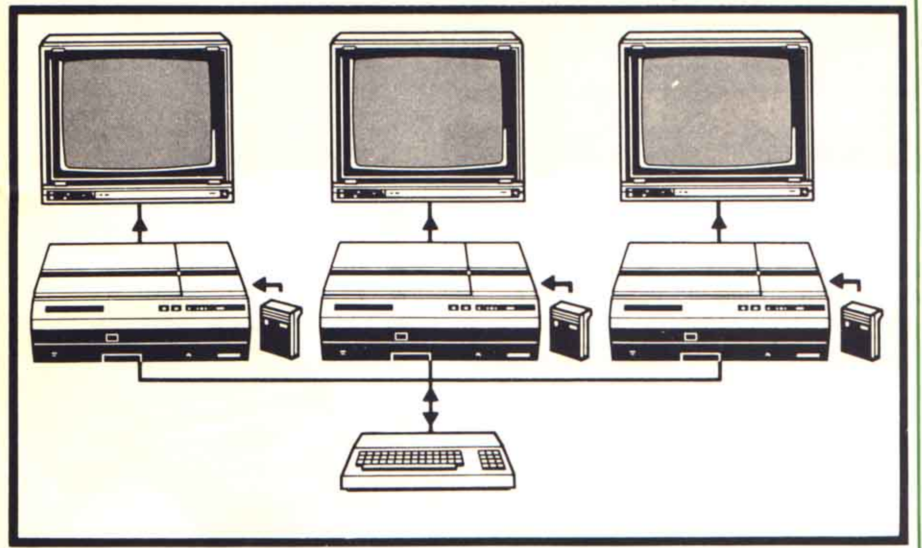


Foto 2. Una de las posibles conexiones del ordenador y las unidades de Video Disco.

Tecnicismos

Un disco *CD-ROM* almacena 270,000 bloques de datos; cada uno de los cuales 2048 contiene *bytes* de datos de usuario, más un número determinado de *bits* de sincronismos y cabeceras. A los datos sólo se puede acceder en forma de uno o varios bloques. Cada bloque de datos en el disco tiene en el bloque de la cabecera la dirección abasoluta de tiempo. La cabecera comprime los cuatro *bytes*, cada uno de los cuales almacena dos dígitos en *BCD* (Binary Coded Decimal-Decimal Coificado en Binario) para lsos «minutos», «segundos», «número de bloque» y «modo». Existen 75 bloques de 2048 *bytes por cada segundo en un disco de 60 minutos.*

Un rayo láser se encarga de ir leyendo la información, evitando que el disco se deteriore, al no tocar la superficie de este.

Ni la versión de Phlips, ni la de *Sony*,tratan los *Compact Disc* como las unidades de *diskettes*. Ambos utilizan los circuitos de interface muy simples. A los datos que accede mandando una instrucción al registro de control de instrucciones que especifican el minuto, segundo, número de bloque y modo del párametro; en caso del aparato *Sony*, el modo del parámetro será o un número (indicando el mismo de bloques a leer) ó cero (indicando que estos datos deberán ser leídos a partir de un bloque del disco hasta el final de éste).

El *Compac Disc* utiliza un interface en serie, mientras *Sony* ha adoptado un interface en paralelo. *Sony* utiliza ocho líneas bidireccionales de datos: líneas de *DATA/COMMAND* que selecciona los datos y/o los registros de las instrucciones en el aparato; línea para lectura *(READ)*; línea de *COMMAND WRITE* que indica la escritura de instrucciones; línea de *STATUS ENBLE*, que indica cuando se necesita la información; línea *WAIT/DRQ* indicando cuando se han actualizado los datos del registro; cuatro líneas de *DRIVE SELECT* (se pueden conectar hasta cuatro aparatos); una línea *RESET*; y una línea *SELECT WT/DRQ* que trabaja con la línea *WAIR/DRQ*.

Los requerimientos de interface son unos *buffer* TTL, tales como los 74LS14 y 74LS125 en el ordenador.

El juego del código de instrucciones es bastante trivial: A0 significa, busca y lee bloques uno cada vez; A8 siginifica busca y lee los bloques contínuamente y F0, significa *BREAK*. Los códigos de instrucción A0 y A8 tienen cuatro parámetros, que son los «minutos», «números de bloques», y «N» (siendo N el número de bloques a leer).

El estado del sistema se obtiene de una palabra de 8 *bits* en el registro de *status*. El significado de cada *bit* es el siguiente:

BIT	SIGNIFICADO
0	0: fin de lectura
	1: leer bloque siguiente
1	0: no hay error de datos
	1: error de datos
2	0: no hay error de instrucción
	1: error en la instrucción

3 0: no hay error de búsqueda
1: error de búsqueda
4 0: disco correcto
1: modo equivocado del disco en el *Compac Disc*
5 0: disco correcto
1: disco de audio en el *Compac Disc*
6 0: disco correcto
1: disco ilegal en el *Compac Disc*
7 0: disco correcto
1: no hay disco en el *Compac Disc*

La gran capacidad del disco permitirá grabar una completa Base de Datos, con gráficos, sonidos, etc.

la de mantener los dedos cruzados y esperar lo mejor.

La información grabada en *cassette*, es mas propensa a los errores debido principalmente al arrastre de la cinta, a las variaciones en la velocidd y a los ruidos. En este caso la mejor política es la de no preocuparse o utilizar *chechsums* para verificar los datos leídos. El almacenamiento de datos en cassette, normalmente se utiliza en aplicaciones que no son importantes y en ordenadores domésticos, donde los errores no son tan perjudiciales como lo pueden ser en una empresa.

Los datos almacenados digitalmente en cintas de vídeo y los almacenados en *Compact Disc* serían casi inservibles sin las avanzadas

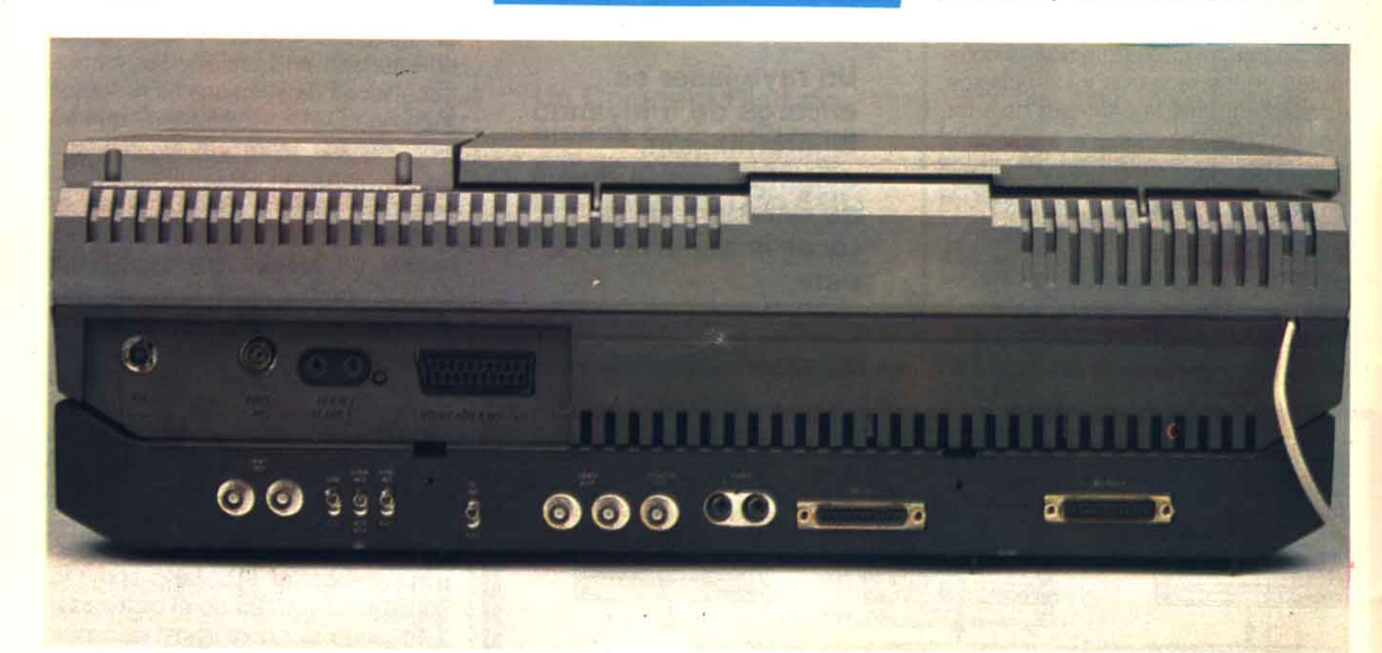


Foto 3. La unidad de Phlips viene provista de euroconector de vídeo/audio, interface RS-232C y conectores de antena de televisión y monitor entre otros.

Corrección de errores

Cuanto más débil sea el medio de almacenamiento, mayor atención habrá que poner para corregir el problema de lectura de datos.

Muchas técnicas de correción de errores han sido desarrollada a través de los datos, variando en complejidad desde la paridad simple hasta los complicados métodos de redundancia. Esta cuestión es demasiado extensa como para tratarla aquí con detenimiento, de forma que lo trataremos brevemente.

Las memorias, particularmente la *ROM*, no son inmunes a los errores. Ocasionalmente, y el *IBM PC* es un caso a tener en cuenta, los fabricantes se preocupan de añadir un *bit* extra de paridad por cada palabra en la memoria *RAM*. Generalmente, la tasa de errores es demadiado baja como para tomar esas precauciones. Los siguientes en la lista de periféricos son los soportes magnéticos, que utilizan técnicas de grabación moduladas por frecuencia. La característica de una unidad de *diskettes* normal es: errores de búsqueda -1×10^{-6}; errores de fácil detención 1×10^{-9} *bits* y errores problemáticos 1×10^{-12} *bits*. Con una tasa de errores de esta magnitud no es necesarios crear códigos correctores de errores, aunque la tónica general es



Foto 4. Con el Vídeo Disco se consigue un sonido en alta fidelidad y unas imágenes clarísimas, además de una potente Base de Datos.

técnicas de correción de errores. El método utilizado con los *CD-ROM* se denomina «redundancia con intervalos».

Cada palabra de datos se graba varias veces, pero no de forma contigua. Las palabras redundantes se reparten por diversas zonas separadas del disco. Supongamos que una burbuja o mota de polvo ha causado un error en la lectura del dato. Puede ocurrir que el motivo del error sea lo suficientemente grave como para alterar todas las palabras redundantes, en caso de que éstas se encuentren lo suficientemente juntas, por este motivo se graban cada una en un lugar del disco. Por este motivo, existe un *buffer* de tamaño lo suficientemente grande como para almacenar un bloque de palabras y efectuar una comprobación de las palabras redundantes antes de pasar la información al ordenador y actualizar el *buffer*.

La unidad de Compact Disc de Phlips utiliza un interface serie, mientras que la de Sony utiliza un interface paralelo.

Conclusión

La tecnología del *Compac Disc* hará que el usuario del ordenador personal posea un periférico que le permita disfrutar el almacenamiento masivo de información que normalmente acompaña a los ordenadores de alto nivel, y a un precio muy asequible, lo que puede traer multitud de consecuencias.

Agradecemos las ayudas prestadas por *Philips* y *Pioneer* sin cuya colaboración no hubiera sido posible la realización de este artículo.

SOFTWARE

Programa: Grand National
Tipo: Juego
Distribuidor: Microbyte
Formato: Cassette

El mundo del caballo, aún hoy día constituye una pequeña élite de personas que se mueven entorno a él, y ocupa cada día más espacio en las páginas deportivas de nuestros diarios.

Pero existen otras personas que son las que verdaderamente hacen que el deporte del caballo sea hoy lo que es.

Las grandes carreras de caballos mueven entre el bullicio, las apuestas y la ilusión, gran cantidad de personas que cada día trabajan para que todo se encuentre en perfecto estado y nada falle.

Otros por el contrario sueñan con poder ir a ver una carrera de gran envergadura o llegar a ser uno de los más grandes «jockeys» mundiales. Este es el caso del «jockey» Bob Champion, el cual después de haber vivido desde pequeño entre caballos y ser uno de los más famosos jinetes tuvo una enfermedad que lo tuvo retirado del deporte durante mucho tiempo, pero tras someterse a un tratamiento químico consiguió volver a montar a caballo y ganar el *Gran National*.

En honor a este gran jinete nosotros seremos esta vez él y tendremos que ganar el el *Gran National*. La carrera consta de 7.220 metros que tendremos que correr en el menor tiempo posible.

También encontaremos a lo largo de nuestra carrera numerosos obstáculos que nos harán caer, pero esto será la menor de las veces ya que nuestra habilidad como corredores se verá reflejada en la carrera.

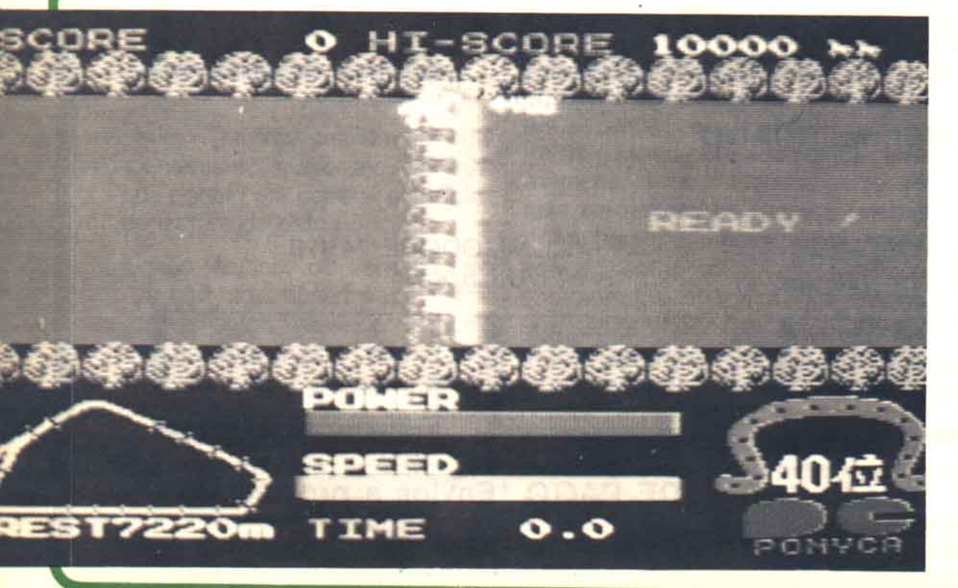
Las reglas del juego son muy fáciles, tendremos que elegir primero la puerta por la que queremos salir, para saber cuál es nuestra posicón.

Utilizaremos nuestro *joystick*, o las teclas de nuestro ordenador, cursores derecho e izquierdo para hacer que el caballo corra. Contaremos también con la posibilidad de hacer que el caballo corra más cada vez que pulsemos el espacio, ya que nos servirá el látigo. Cuando el caballo vaya a saltar uno de los obstáculos, será también la barra espaciadora la que nos ayude a afectuar el salto, pero aquí tendremos que medir muy bien las distancias ya que solamente puntuaremos cada vez que el cabalo salte perfectamente y con la limpieza de los grandes campeones que sin duda somos nosotros.

Los obstáculos se presentan como zanjas, vallas y pequeños tramos con agua que no debemos pisar.

Es un juego especialmente dedicado para los amantes de las carreras de caballos, pero sin duda nos hará pasar un rato entretenido y sentirnos campeones de uno de los grandes acontecimientos deportivos que más riesgo y belleza conlleva, el *Grand National*.

Puntuación:
Presentación: 6
Claridad: 6
Rapidez: 7
Adicción: 7



Programa: Super Cobra
Tipo: Juego
Distribuidor: Serma
Formato: Cartucho

Konami's siempre ha sido un innovador, aunque muchas veces haya trabajado temas clásicos, pero siempre dándoles un matiz que les ha hecho netamente suyos. Este juego para seguir la trayectoria de los cambios, sin duda para mejor, es totalmente nuevo, es un paso más en el complejo, entretenido y macromundo de los juegos para ordenador.

Super Cobra no es un juego de animales, aunque como su nombre indica tenemos que emplear la sutilidad, rapidez e inteligencia de una cobra, para salir al paso de los numerosos problemas que a lo largo del juego se nos plantearán. *Super*

Cobra será el nombre de nuestro helicóptero, el cual se moverá a lo largo de una serie de pantallas, para ser más exactos cuatro. En la primera zona, llamada de los mutantes, seremos asaltados inesperadamente por unos misiles que vendrán tanto del territorio que sobrevolamos como del espacio que surcamos. Tendremos que esquivarlos para poder continuar por otras pantallas.

Los elementos con los que contamos son: capacidad de disparo en sentido horizontal del helicóptero, y a su vez, poder dejar caer bombas sobre los blancos situados sobre el territorio. El helicóptero llevará una trayectoria uniforme en velocidad, por tanto sólo podremos movernos hacia arriba o hacia abajo, y si queremos frenar moveremos nuestros cursores o palanca del joysticks hacia atrás.



Las siguientes zonas por las que pasaremos si conseguimos salir sanos de las anteriores, serán una segunda zona llamada de los «ovnis», en la que además de contar con el fuego de la anterior pantalla, aparecerán multitud de ovnis que nos atacarán sin piedad y más de una vez nos destruirán.

La tercera zona, por si usted es demasiado hábil y considera que el juego no tiene emoción, se internará en una gruta, si, sí, tendrá que sortear no sólo las balas enemigas y todo aquél objeto que tenga como fin destruirle, sino que además deberá intentar no chocarse con las estalactitas o trozos de roca que forman el techo de la gruta.

No se preocupe si le destruyen una o dos veces, ya que contará con tres helicópteros a lo largo de la partida, pero no los desprecie.

También es necesario que dispare a los tanques de combustible que encuentre, puesto que así conseguirá gasolina. Si se le acaba caerá rápidamente al suelo y finalizará la partida.

En la cuarta zona no sólo tendra que esquivar el peligro, sino que tendrá que ayudar a su ejército al transporte de..., le dejaremos que lo descubra por sí mismo.

Puntuación:
Presentación: 8
Claridad: 8
Rapidez: 7
Adicción: 7

Programa: Ping-Pong
Tipo: Juego
Distribuidor: Serma
Formato: Cartucho

El tenis de mesa más conocido como *ping-pong*, es un deporte relativamente moderno, se cree que data en 1880, aunque no se posee datos concretos al respecto. Su origen no se encuentra como muchos creen en el Japón, sino que existe una gran discusión entre americanos e ingleses.

No obstante, debemos reconocer a los japonses como verdaderos descubridores de este juego, ya que la práctica habitual tanto de niños como de personas mayores les ha hecho ponerse a la cabeza del *ping-pong* mundial. No sólo han conseguido proclamarse campeones mundiales, sino que se les considera como los grandes maestros de este deporte, por lo que indidublemente el maestro de los juegos es el ordenador.

El deporte del *ping-pong* que hoy comentamos, será para los aficionados a este deporte y para aquellos que nunca han jugado, una diversión sin límite ni horas.

Refiriéndonos concretamente a nuestro *Konami's Ping-Pong*, vamos a describir a través de él las características de este deporte, su desarrollo, elementos que se utilizan y el escenario, fundamental en cualquier encuentro deportivo. Comenzaremos presentando los elementos, una mesa en cuyo centro se eleva una red de 22 cm., la cual divide los campos de juego.

Cada adversario contará con una pala o raqueta con la que intentará golpear la pelota lo más hábilmente que sepa, para conseguir efectos, bloqueos, e incluso si nuestra habilidad es más extrema, realizar un dificultoso *top spin* (pegada con efecto), que sin duda nos erigirá como campeones de este deporte. Todos los elementos y sin duda muchos más están claramente dispuestos en el *ping-pong* de Konami porque jugar a este deporte no es sólo intentar dar a la pelota, sino que exige inmejorable forma física y un nivel de concentración máximo, ya que usted podrá comprobar tanto en el juego como en la realidad, la velocidad y la precisión de movimientos con que se desarrolla el juego

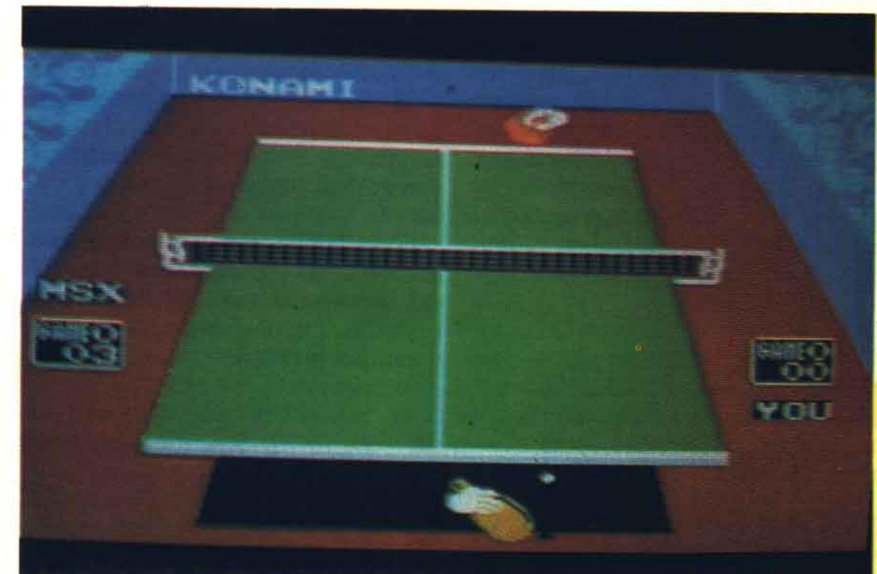
Konami's ante todas estas características, ha diseñado un juego para que todo el mundo pueda divertirse con él y pasar un buen rato de nervios. El juego tiene cinco niveles que van desde el más fácil para principiantes hasta el más complicado para verdaderos maestros, en este último aquellos que verdaderamente sean fanáticos de este deporte verán consumadas sus aspiraciones.

Nos moveremos con nuestros cursores y joystick a lo ancho de la mesa, teniendo la posibilidad de

efectuar golpes de revés con sólo pulsar al mismo tiempo la barra espaciadora. Tendremos a derecha e izquierda de nuestra pantalla los marcadores que irán diciendo los puntos que conseguimos y un público que nos motivará apludiéndonos, pero hay que ser muy rápidos.

El sonido es otro elemento a destacar. Debido al nivel de concentración necesario, sólo se notará cuando hayamos conseguido un punto, o finalice el partido.

Si verdaderamente sabemos apreciar las características de este juego, quizás algún día logremos enfrentarnos a los más consabidos profesionales, habiendo recibido antes unas «pequeñas lecciones».



Puntuación:
Presentación: 9
Claridad: 8
Rapidez: 9
Adiccion: 9

COGE EL X'PRESS

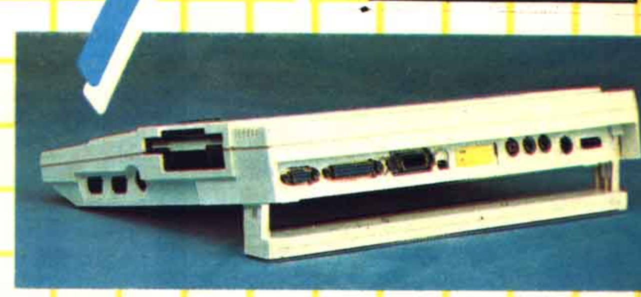


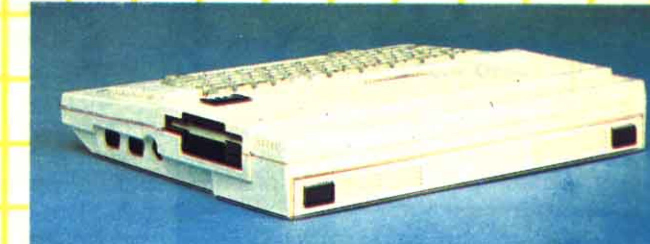
99.900 pts

SVI·738



- Ordenador Sistema MSX
- 80K RAM
- Unidad de disco de 3,5" integrada en el teclado
- Trabaja en CP/M, MSX—DOS, MSX—DISK BASIC
- Teclado profesional de diseño ergonómico. Va incluído un maletín para la protección del ordenador durante su transporte
- Dos puertas de conexión: RS 232—C y Centronics paralelo
- Salida a T.V. y monitor
- Admite directamente una segunda unidad de disco
- MVDP (pasa de 40 a 80 columnas en pantalla. Indispensable para trabajar en CP/M)









SOFTWARE

Programa: Phile
Tipo: Aplicación
Distribuidor: Philips
Formato: Cassette

Todos aquellos amantes de un orden y una diversificación de tareas, y en general para todos aquellos que les gusta llevar controlada su economía tanto doméstica o de ámbito general (funciones propias de este programa), han encontrado sin duda en esta base de datos una gran capacidad para todas las necesidades de control y orden del usuario de MSX.

Tan solo encontraremos un inconveniente, que no es tal, en el momento de grabar datos o cargarlos, ya que el soporte de almacenamiento, al ser secuencial, es lento.

Las funciones del programa PHIFILE, están divididas en varios apartados o menús. El primero que nos aparece lo denominaremos «menú principal», el cual engloba:

1. Proceso del fichero.
2. Procesos con fichas.
3. Información del fichero.
4. Definir formato de impresión.
5. Terminar la sesión.

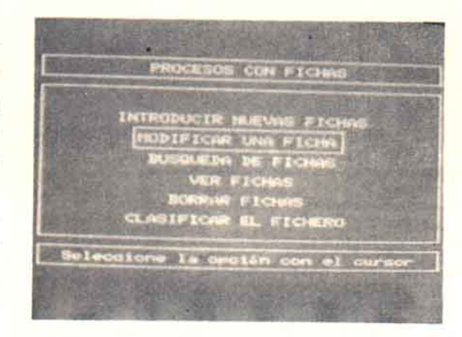
Utilizaremos nuestros cursores inferior y superior para colocarnos en una de las opciones deseadas, y pulsaremos la tecla «enter» o «return», así la opción elegida nos dará paso a un nuevo menú de opciones que cada una de ellas engloba.

Vamos a ver cada una de ls opciones del menú principal, los apartados y nuevas opciones que nos presenta:

1. Proceso del fichero: Aquí se desarrollan todas las opciones referentes a la gestión de ficheros, como son: «Crear ficheros», donde definiremos el fichero que vamos a utilizar para que el ordenador sepa como queremos la información. «Cargar un fichero», si no es la primera vez que trabajamos con un fichero, esta opción nos permitirá cargarlo en memoria y así poder utilizarlo luego.

El ordenador nos pedirá el nombre del fichero, que ha de ser de ocho letras, y aparecerá a continuación, el mensaje «Pulse otra tecla», para así darnos tiempo a preparar la cinta en nuestro cassette.

Pulsándo cualquier tecla se cargará el fichero deseado, salvo si pulsamos la tecla «ESC», con la que retrocederemos al menú principal.



«Grabar fichero», que como propiamente indica nos posibilitará directamente grabar en nuestro cassette.

«Compactar fichero», nos servirá para que aquellas fichas anuladas del fichero no nos ocupen espacio y así lograr que éste sea más pequeño.

2. Procesos con fichas: Este apartado hace referencia a: «Introducir nuevas fichas», una vez creado el fichero, es necesario introducir las fichas que van a conformar la información de nuestro fichero.

«Modificar una ficha», nos pedirá el número de la ficha a modificar, e instantáneamente nos aparecerá en la pantalla la ficha pedida, y el cursor sobre el primer campo para modificar aquello que sea necesario.

«Búsqueda de fichas», cuando deseemos seleccionar una o más fichas, que cumplan una condición dada, accederemos a este apartado, que permite buscar fichas en el archivo, estableciendo un criterio de selección. En la línea inferior aparecerán las diferentes posibilidades de selección: mayores que, menores que, distintas que, etc.

«Clasificar el fichero», esta opción ordena las fichas, según el orden establecido, de uno de los campos de las fichas. La clasificación se efectuará a tenor de las opciones: de menor a mayor o bien, de mayor a menor.

elegido el criterio de clasificación, sale el formato de la ficha en la pantalla, debiendo situar el cursor sobre el campo por el que deseemos ordenar el fichero.

3. Información del fichero: Este proceso muestra la información acerca del fichero que esté cargado en memoria. Los datos que aparecen en pantalla son: Formato de la ficha, nombre del fichero, y número de fichas que contiene. Así, pulsando la tecla (que no sea «ESC», con la que regresaremos al menú principal) obtendremos.

— N.° de campos por ficha.
— Memoria disponible.

4. Definir formato de impresión: Determinaremos aquí el orden con el que queremos que aparezcan impresos los campos en una ficha. Por tanto si una ficha consta de tres campos, como pueden ser, dirección y teléfono, y les numeramos 2, 1, 3, nos parecerá el formato de la ficha de la siguiente forma:

— Dirección, nombre y teléfono.

Si deseamos que uno de los campos a la hora de imprimirlo no aparezca, basta con no numerarlo.

Es una base de datos sin duda imprescindible para el usuario de MSX, no obstante no hemos enumerado todas las posibilidades del programa, que esperemos usted descubra.

Puntuación:
Presentación: 7
Claridad: 8
Rapidez: 8

Programa: Fruit Panic
Tipo: Juego
Distribuidor: Microbyte
Formato: Cassette

Una vez más, volvemos a dedicar nuestro comentario a un juego de características parecidas al «comecocos», pero *Fruit Panic»* es algo más, representa la renovación de forma y la variación de contenido en un juego que nos será difícil encontrar su significado, pero no obstante han sido los pioneros de los juegos informáticos y aún hoy en día son indispensables.

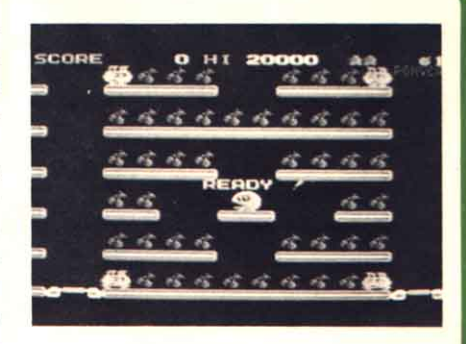
Muchas personas se habrán preguntado qué es lo que encontramos en ese tipo de juegos para no dejar de utilizarlos, la respuesta es muy sencilla: entretenimiento. Debemos reconocer que los antiguos «comecocos» han sido superados, nuestro fin ya no será el ir comiendo «bolitas» mientras que pequeños fantasmas intentan alcanzarnos. Desarrollándose todo en un clima de sencillez y con el tiempo a nuestro favor, tenemos ante nosotros un prototipo que engloba todas esas características, el *Fruit Panic*.

Walky, que va a ser nuestro amigo, será el compañero con el que contaremos para comer el mayor número de frutas a lo largo de la pantalla. Tendremos por el contrario a los temibles *Ñankies*, que intentarán comernos sin piedad.

Pero *Walky*, *Ñanky*, frutas, ¿de qué están hablando? No se preocupen,sólo es una pequeña introducción. El juego se desarrolla en un almacén de frutas, las cuales se encuentran acumuladas en filas, las cuales deben ser comidas lo más rápidamente posible por nuestro amigo *Walky*, ya que los guardianes o *Ñankies* nos pueden comer. El tiempo, que en este caso también es un enemigo, nos hará pasar malas jugadas.



El sistema de juego es muy sencillo, tendremos que comer un número de frutas de la misma clase —que nos supondrá una puntuación— sin dejarnos atrapar. Utilizaremos unos resortes para poder subir de una tabla a otra. Estos pequeños trampolines son tan sólo para el uso de los *Ñankies*, por tanto, si nos paramos en ellos o saltamos cuando estén en una tonalidad roja perderemos un *Walky*. Contaremos con elementos de defensa como son unas pequeñas bombitas que dejarán adormilados por un rato a nuestros perseguidores. Conseguiremos obtener más



bombas a medida que consigamos mayor número de puntos.

Son nueve pantallas con sus frutas correspondientes, encontraremos desde cerezas hasta sandías, pasando por estrellas, que no será una pantalla de frutas, sino de récord ya que esta vez unicamente el tiempo será nuestro enemigo. Si conseguimos acabar toda la pantalla antes de que pase el tiempo subiremos puntos, pero tambien las nuevas pantallas serán más difíciles, ya que la velocidad de los *Ñankies* se irá incrementando.

Es un juego entretenido que encabeza una nueva manera de ver los «come frutas» ya que la singularidad del juego no nos permitirá englobarlo dentro de nuestros antiguos juegos.

Puntuación:
Presentación: 7
Claridad: 7
Rapidez: 7
Adiccion: 9



el Dynadata DPC-200



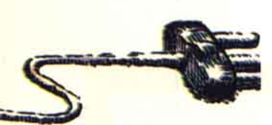
En esta sección que estrenamos este mes, vamos a introducir al lector los ordenadores y periféricos que, en sucesivos meses vayan saliendo al mercado.

Un nuevo ordenador se incorpora al ya abundante mercado de ordenadores personales MSX. Bajo el nombre de Dynadata DPC-200, fabricado por Dynadata, se introducirá en nuestro país este aparato, cuyas características principales son las mismas que las del resto de la gama MSX.

No vamos a comentar sus interioridades, puesto que verifica todo lo ya expuesto y comentado acerca del estándar.

A primera vista el aparato presenta un aspecto elegante. Sobre la carcasa de color blanco, contraste el bloque de teclas de control del cursor y una cubierta de plástico negro que protege el *slot*para la conexión de cartuchos. El teclado es uno de los puntos fuertes de este ordenador, ya que ofrece una tacto suave y agradable, además de tener grabados en las teclas todos los caracteres gráficos que se pueden obtener, con lo que nos ahorramos una consulta al manual cada vez que necesitemos algún carácter gráfico. La única pega que podemos ponerle a este teclado es que las teclas de función y control, tales como *STOP* o*DEL*, son demasiado pequeñas por lo que resulta fácil equivocarse.

Las salidas de vídeo, audio y RF modulada para la conexión al televisor, así como la salida para impresora y el bus de expansión del sistema, se encuentran situadas en la parte trasera del aparato. Las conexiones del cassette y joysticks se encuentran en el lateral derecho del aparato, lugar muy accesible para estas conexiones ya que son las más utilizadas. En la parte superior del aparato se encuentra un *bus* de expansión (réplica el que hay en la parte posterior)



protegido por una tapa de plástico, además de poseer otro sistema de protección, un interruptor que desconecta el equipo cada vez que se introduce algún cartucho, evitando así introducir cartuchos *ROM* cuando el ordenador se encuentra encendido, lo que puede causar serios problemas.



Figura 1. Las conexiones estándar están bien estudiadas.



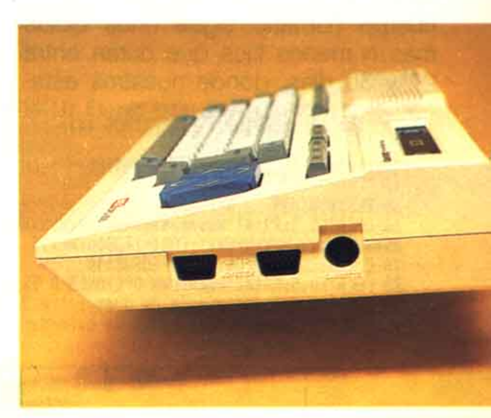
Además de los conectores imprescindibles, se suministran con el ordenador dos cables adicionales para la conexión de éste a un amplificador externo y a un monitor, todos ellos de buena calidad.

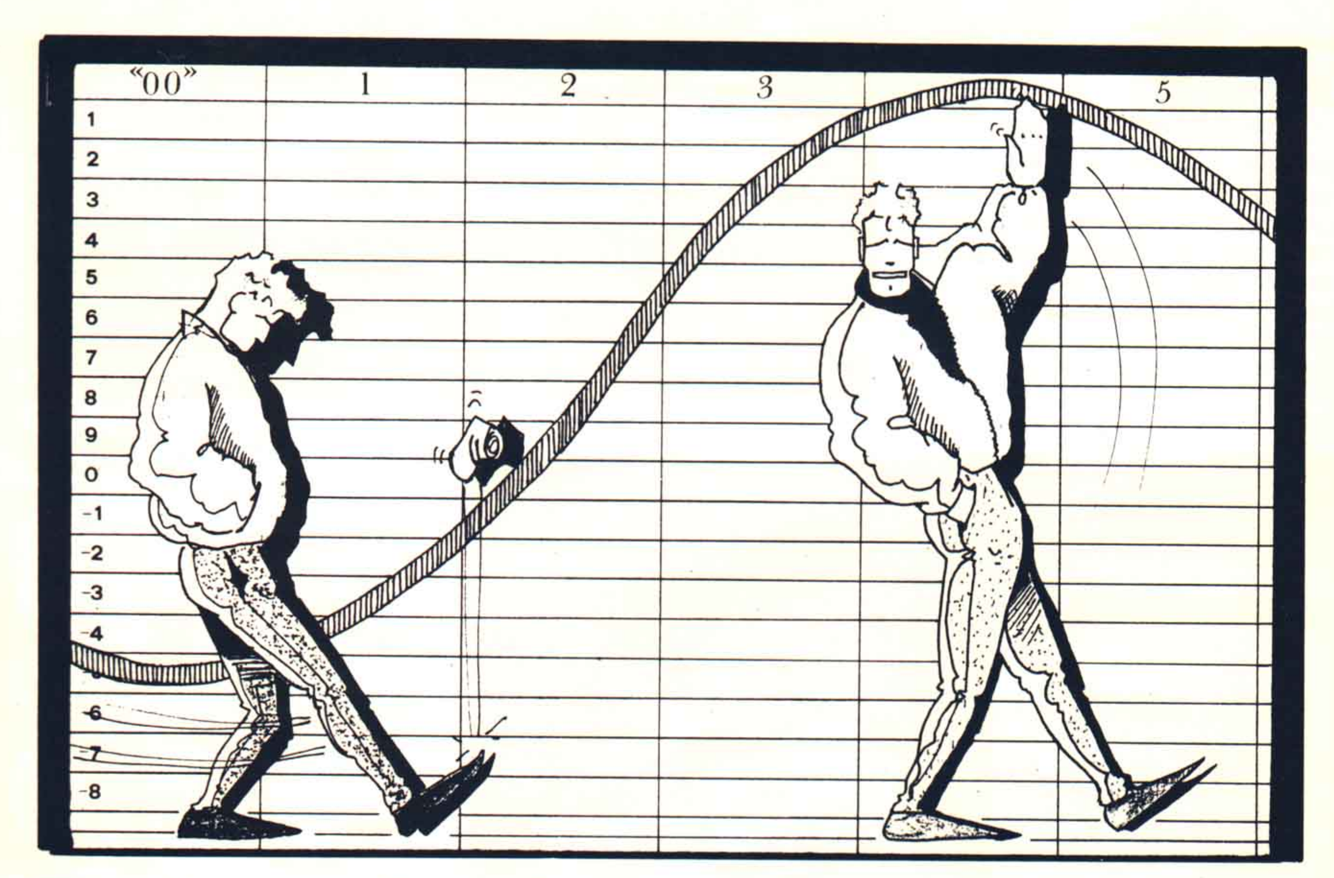
El manual que acompaña al ordenador está dividido en dos partes. La primera de ellas explica claramente la conexión y puesta en marcha del ordenador, mientras que en la segunda parte se nos introduce de una forma rápida y sencilla al BASIC MSX para terminar con una descripción de todas las palabras clave de este lenguaje. Completando este manual encontramos una cinta con tres programas de demostración: un programa para la creación de sonidos, otra para la generación de gráficos y, como no, un juego.

Este ordenador no tiene nada que envidiar a la mayoría de los ya existentes MSX en el mercado, llegando incluso a superar a algunos de estos.

Con un precio de 49.400 ptas. se impone como la opción más barata del mercado.

Figura 2. Su teclado y su precio, son sus mejores armas.





Calendario y biorritmos

No todos los días nos levantamos con el mismo humor ni con las mismas ganas. A veces los lunes se nos convierten en el principio de un calvario que durará cinco días, otras veces estamos deseando que llegue el lunes para comernos el mundo, ¿a qué se debe esta diferencia en el carácter de cada uno? Sabemos que el cuerpo humano sigue unos ciclos más o menos fijos que duran entre 27 y 30 días, donde nuestros estados anímicos, emotivo, físico e intelectual, atraviesan unos máximos y unos mínimos.

Cuando estamos en un ciclo donde los tres apartados anteriores están al máximo, nos veremos en la situación de querer acabar con todo el mundo; como se diría vulgarmente, estaríamos a tope, mientras que cuando estemos en los tres apartados al mínimo, lo mejor que podemos hacer es no levantarnos de la cama hasta que se nos pase.

Conozca su estado de ánimo con el programa siguiente y si no le gusta lo que ve, dedíquese a obtener los calendarios de años venideros.

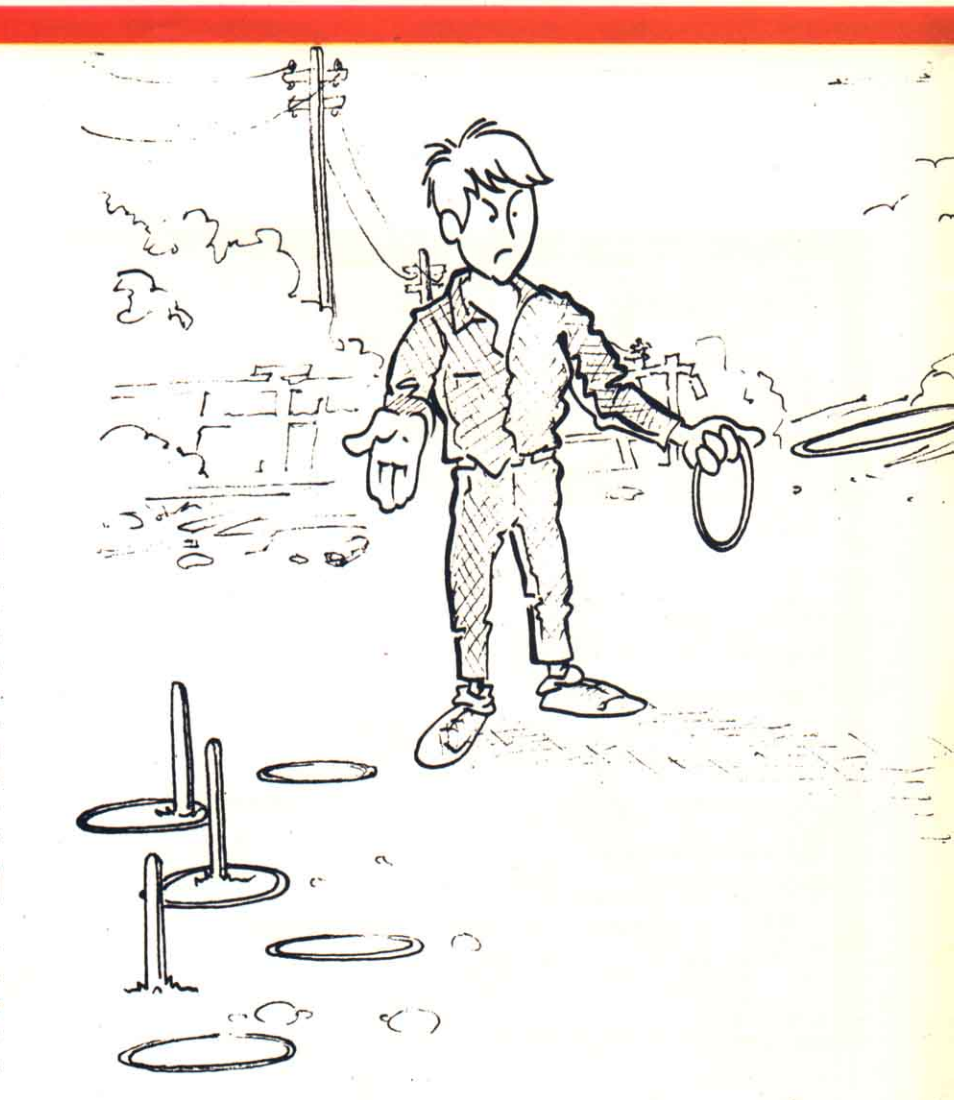
El programa, originalmente es para el SV-328, pero los usuarios de MSX podrán utilizarlo alterando las líneas debidas.

José Closas Compte
Barcelona

```
10 '*** CALENDARIOS Y BIORRITMOS ***
20 '
30 '  Jose Closas Compte  4-4-1985
40 '
50 CLS:WIDTH39
60 LOCATE0,5:PRINT"WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW":LOCATE6,10:PRINT"CALENDARIOS Y BIORRITMOS":LOCATE0,15:PRINT"WWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWW":FORT=1TO800:NEXTT
70 CLEAR:CLS:STOPON:ONSTOPGOSUB640
80 LOCATE0,5:PRINT"WWWWWWWW OPCIONES A ESCOGER: WWWWWWWW"
90 LOCATE2,7:PRINT"N(mero de d)as entre 2 fechas .. 1"
100 LOCATE2,9:PRINT"Impresi\n de calendarios ....... 2"
```

```
110 LOCATE2,11:PRINT"Biorr\tmo semi gr/fico ......... 3"
120 LOCATE0,13:PRINT"WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW"
130 LOCATE2,16:PRINT"Entre la opci\n deseada:"
140 STOPON:ONSTOPGOSUB640
150 A$=INKEY$:IFA$="1"THEN190
160 IFA$="2"THEN360
170 IFA$="3"THEN650
180 GOTO150
190 CLS:LOCATE8,8:PRINT"Fecha 1:"
200 LOCATE17,8:INPUT"a+o";AA:IFAA<0THENBEEP:GOTO200
210 LOCATE17,10:INPUT"mes";MA:IFMA<1ORMA>12THENBEEP:GOTO210
220 LOCATE17,12:INPUT"d\a";DA
230 IFMA=1ORMA=3ORMA=5ORMA=7ORMA=8ORMA=10ORMA=12THENGOSUB540:GOTO260
240 IFMA=4ORMA=6ORMA=9ORMA=11THENGOSUB550:GOTO260
250 GOSUB560
260 IFDA<1ORDA>NTHENBEEP:GOTO220
270 CLS:LOCATE8,8:PRINT"Fecha 2:"
280 LOCATE17,8:INPUT"a+o";AD:IFAD<0THENBEEP:GOTO280
290 LOCATE17,10:INPUT"mes";MD:IFMD<1ORMD>12THENBEEP:GOTO290
300 LOCATE17,12:INPUT"d\a";DD
310 IFMD=1ORMD=3ORMD=5ORMD=7ORMD=8ORMD=10ORMD=12THENGOSUB540:GOTO340
320 IFMD=4ORMD=6ORMD=9ORMD=11THENGOSUB550:GOTO340
330 GOSUB580
340 IFDD<1ORDD>NTHENBEEP:GOTO300
350 GOSUB450:CLS:LOCATE8,8:PRINT"N(mero de d\as=";D:GOTO600
360 CLS:LOCATE8,8:INPUT"A+o deseado";AD:IFAD<0THENBEEP:GOTO360
370 DA=1:MA=1:AA=1990:DD=1:MD=1:GOSUB450:IFAA>ADTHEND=-D
380 S=INT(D/7):R=D-S*7:WIDTH21:PRINT"  CALENDARIO DE";AD:PRINT
390 FORI=1TO12:READMES$:AUX=INT(LEN(MES$)/2):PRINTTAB(11-AUX);MES$:PRINT:PRINT" LU MA MI JU VI SA DO":PRINTTAB(3*R);USING"###";1;:ONIGOS
UB540,580,540,550,540,550,540,540,550,540,550,540
400 CO=R+1:IFCO=7THENCO=0:PRINT
410 FORJ=2TON:PRINTUSING"###";J;:CO=CO+1:IFCO=7THENCO=0:PRINT
420 NEXTJ:PRINT:PRINT:R=R+N-INT((R+N)/7)*7
430 NEXTI:FORT=1TO800:NEXTT:WIDTH39:LOCATE0,7:GOTO600
440 DATAENERO,FEBRERO,MARZO,ABRIL,MAYO,JUNIO,JULIO,AGOSTO,SEPTIEMBRE,OCTUBRE,NOVIEMBRE,DICIEMBRE
450 A1=INT(ABS((AA+3)*365.25))
460 A2=INT(ABS((AD+3)*365.25)):IFMA=1THEN490
470 FORI=1TOMA-1:ONIGOSUB540,560,540,550,540,550,540,540,550,540,550,540
480 C1=C1+N:NEXTI
490 IFMD=1THEN520
500 FORI=1TOMD-1:ONIGOSUB540,580,540,550,540,550,540,540,550,540,550,540
510 C2=C2+N:NEXTI
520 D1=C1+A1+DA:D2=C2+A2+DD
530 D=ABS(D1-D2):RETURN
540 N=31:RETURN
550 N=30:RETURN
560 N=28:IFINT(AA/4)=AA/4THENN=29
570 RETURN
580 N=28:IFINT(AD/4)=AD/4THENN=29
590 RETURN
600 PRINT:PRINT:PRINTTAB(5);"Quiere volver al men(?(S/N)
610 B$=INKEY$:IFB$="S"ORB$="s"THEN70
620 IFB$="N"ORB$="n"THENCLS:END
630 GOTO610
640 WIDTH39:CLS:END
650 CLS:LOCATE8,8:PRINT"Fecha de nacimiento:"
660 LOCATE8,10:INPUT"a+o";AA:IFAA<0THENBEEP:GOTO660
670 LOCATE8,12:INPUT"mes";MA:IFMA<1ORMA>12THENBEEP:GOTO670
680 LOCATE8,14:INPUT"d\a";DA
690 IFMA=1ORMA=3ORMA=5ORMA=7ORMA=8ORMA=10ORMA=12THENGOSUB540:GOTO720
700 IFMA=4ORMA=6ORMA=9ORMA=11THENGOSUB550:GOTO720
710 GOSUB560
720 IFDA<1ORDA>NTHENBEEP:GOTO680
730 CLS:LOCATE8,8:PRINT"Fecha deseada:"
740 LOCATE8,10:INPUT"a+o";AD:IFAD<0ORAD<AATHENBEEP:GOTO740
750 LOCATE8,12:INPUT"mes";MD:IFMD<1ORMD>12THENBEEP:GOTO750
760 CLS:LOCATE8,8:PRINT"F: Estado f\sico.":LOCATE8,10:PRINT"E: Estado emotivo.":LOCATE8,12:PRINT"I: Estado intelectual.":FORT=1TO1000:NE
XTT
770 DD=1:GOSUB450:E1=D-INT(D/23)*23:E2=D-INT(D/28)*28:E3=D-INT(D/33)*33:PI=3.141592654#
780 WIDTH25:FORI=1TOMD:READMES$:NEXTI:PRINT"  FECHA.DE NACIMIENTO:":PRINT:PRINTTAB(5);DA;"-";MA;"-";AA:PRINT:PRINTTAB(8-INT(LEN(MES$)/2)
);MES$;" DE";AD:PRINT
790 A$="I---------+---------I":B$="I         +         I":PRINTTAB(5);"-    0    +":PRINTA$:ONMDGOSUB540,580,540,550,540,550,540,540,550
,540,550,540:FORK=1TON:L=0:IFK=5*INT(K/5)THENL=1
800 IFL=1THENC$=A$ELSEC$=B$
810 G=SIN(((K+E1)/23*360)*PI/180):P$="F":GOSUB840
820 G=SIN(((K+E2)/28*360)*PI/180):P$="E":GOSUB840:G=SIN(((K+E3)/33*360)*PI/180):P$="I":GOSUB840
830 PRINTC$;USING"###";K:NEXTK:PRINTA$:FORT=1TO800:NEXTT:WIDTH39:LOCATE0,7:GOTO600
840 M=11+INT(8*ABS(G))*SGN(G):C$=LEFT$(C$,M-1)+P$+RIGHT$(C$,21-M):RETURN
```

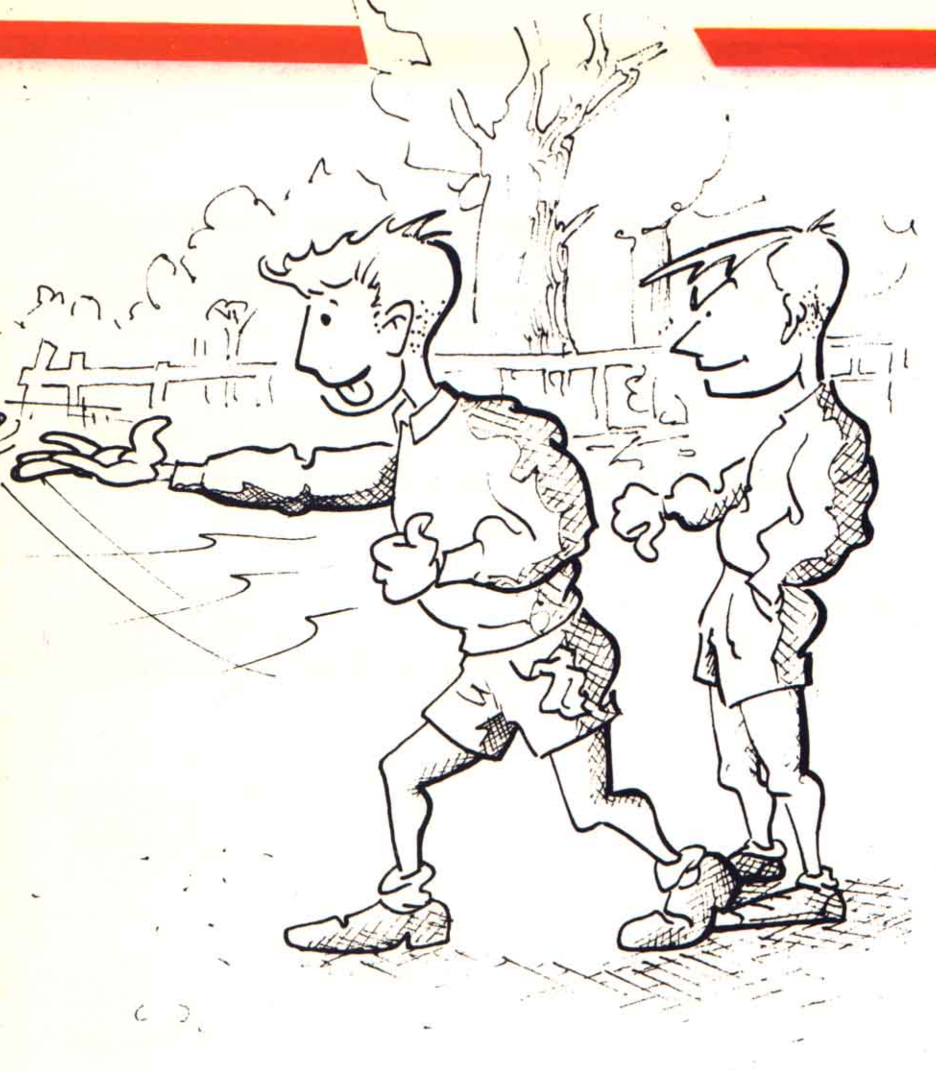
Toka



Es una versión computerizada de un popular juego, del que habremos disfrutado más de una vez. Se trata de lanzar un aro e introducirlo en una barra vertical, situada a cierta distancia. Nuestro aro es metálico y su peso engaña, ya que cuando menos lo esperamos estamos lanzándolo a una distancia muy superior a la necesaria. Para ello, tienes que pulsar las teclas 0 y 9, teniendo en cuenta que la distancia a la que caiga el aro depende de la velocidad con la que se hayan pulsado dichas teclas. Cuanto más rápido se haga, más lejos irá el aro y viceversa.

El semáforo que aparece te indicará cuando puedes realizar el lanzamiento. Además, también cuentas con dos posibilidades, una para entrenarte y otra para competir. En la

```
10 CLS:KEYOFF
20 COLOR15,12,1
30 LOCATE12,2:PRINT"**** TOKA ****"
40 PRINT:PRINT
50 PRINT" El juego de la TOKA consiste en lan-zar unas fichas metálicas contra u
na barra,también metálica."
60 PRINT:PRINT" En la versión para ordenador se lan-zan las fichas pulsando,segu
idas,las teclas [9] y [0].Cuanto más tiempo se deje pasar entre una y otra,más c
er-ca caerá la ficha."
70 PRINT:PRINT" No es fácil;si ves que la ficha cae demasiado cerca,tira más ráp
ido,y másdespacio si cae demasiado lejos."
80 PRINT:PRINT:PRINT:PRINT:PRINT"---------- PULSA UNA TECLA ----------"
90 IFINKEY$=""THEN90
100 CLS
110 LOCATE0,3:PRINT" Para tirar espera que la luz del se-máforo se ponga verde"
120 LOCATE0,6:PRINT" Si quieres parar el juego,aprieta latecla [F1]."
130 LOCATE0,9:PRINT" Al final del juego aparece un marca-dor con el tanteo y el
porcentaje de acierto"
140 LOCATE2,13:PRINT" Para jugar tienes dos opciones:"
150 LOCATE2,15:PRINT"1.-ENTRENAMIENTO.-No tienes número límite de tiradas."
160 LOCATE2,18:PRINT"2.-COMPETICION.-Para dos jugadores;seis tiradas cada uno."
170 PRINT:INPUT"OPCION (1-2)";OP$
180 IFOP$="1"THENP=85ELSEP=71
190 SCREEN2,2
200 COLOR15,7,1
210 LINE(0,0)-(255,191),7,BF
220 OPEN"GRP:"AS#1
```



primera podrás tirar las veces que quieras e intentar coger el punto ideal para el lanzamiento, (acertar no es muy difícil pero hay que poner empeño). En la segunda opción te enfrentarás a otro jugador, al que tendrás que superar en seis tiradas.

Al finalizar el juego, en la primera opción aparecerá un cuadro completo indicando el número de lanzamiento efectuados, los aciertos y el porcentaje. Al finalizar la segunda opción, después de lanzar seis veces cada jugador, se nos mostrará el mismo cuadro con el correspondiente número de aciertos y porcentajes.

Juan María Ruíz Alberca
Irún

```
230 FORI=1TO6:READA:A$=A$+CHR$(A):NEXTI
240 SPRITE$(4)=A$
250 FORI=1TO7:READA:B$=B$+CHR$(A):NEXTI
260 SPRITE$(5)=B$
270 FORI=1TO16:READA:C$=C$+CHR$(A):NEXTI
280 FORI=1TO16:READA:D$=D$+CHR$(A):NEXTI
290 SPRITE$(6)=C$+D$
300 FORI=1TO15:READA:E$=E$+CHR$(A):NEXTI
310 SPRITE$(7)=F$
320 FORI=1TO2:READA:F$=F$+CHR$(A):NEXTI
330 SPRITE$(8)=F$
340 SPRITE$(9)=CHR$(252)
350 FORI=1TO15:READA:G$=G$+CHR$(A):NEXTI
360 SPRITE$(3)=C$+G$
370 FORI=1TO15:READA:H$=H$+CHR$(A):NEXTI
380 SPRITE$(2)=C$+H$
390 FORI=1TO15:READA:I$=I$+CHR$(A):NEXTI
400 FORI=1TO9:READA:J$=J$+CHR$(A):NEXTI
410 FORI=1TO15:READA:L$=L$+CHR$(A):NEXTI
420 FORI=1TO15:READA:M$=M$+CHR$(A):NEXTI
430 SPRITE$(12)=L$+M$
440 SPRITE$(11)=J$
450 SPRITE$(1)=C$+I$
460 SPRITE$(10)=CHR$(192)+CHR$(192)
470 PLAY"O5S1M5000L8CC.L16EL8G.L16AL8GG.L16AL8F.L16GL8EE.L16FL8D.L16EL8CO4B.L16A
L4G"
480 PLAY"O5S1M5000L8CC.L16EL8G.L16AL8GG.L16AL8F.L16GL8EE.L16FL8D.L16EL2C"
```

```
490 LINE(0,141)-(255,191),3,BF:DRAW"C12BM10,175E20R200G20L200":PAINT(100,156),12
:DRAW"C12BM6,177E24R204G24L204"
500 CIRCLE(70,190),100,3,1,3:CIRCLE(210,180),120,3,.55,2.42:PAINT(210,140),3
510 PUTSPRITE20,(70,85),12,12:PUTSPRITE21,(100,100),12,12:PUTSPRITE22,(91,80),12
,12:PUTSPRITE23,(32,80),12,12:PUTSPRITE24,(210,45),12,12:PUTSPRITE25,(180,50),12
,12:PUTSPRITE26,(170,60),12,12:PUTSPRITE27,(130,95),12,12
520 IFOP$<>"1"THENPUTSPRITE26,(10,10),7,30
530 CIRCLE(225,35),18,11,,,1.3:PAINT(215,30),11
540 LINE(0,130)-(105,129),14,BF:FOR I=10TO105STEP15:LINE(I,131)-(I+1,140),14,BF:
NEXT
550 LINE(136,130)-(255,129),14,BF:FORI=140TO255STEP15:LINE(I,131)-(I+1,140),14,B
F:NEXT
560 LINE(183,124)-(188,140),15,BF:LINE(228,124)-(233,140),15,BF
570 LINE(163,P)-(253,123),1,BF
580 PSET(168,92),1:PRINT#1,"TIROS"
590 PSET(168,107),1:PRINT#1,"BUENAS"
600 IFOP$<>"1"THENPSET(168,77),1:PRINT#1,"JUGADOR"
610 CIRCLE(91,112),5,1,0,3:CIRCLE(91,120),5,1,3,0:LINE(86,112)-(86,120),1:LINE(9
6,112)-(96,120),1:PAINT(91,112),1:LINE(90,126)-(92,140),15,BF
620 FORI=1TO300:NEXT
630 PUTSPRITE1,(30,123),1,4
640 PUTSPRITE2,(30,126),10,5
650 PUTSPRITE3,(25,133),15,6
660 PUTSPRITE4,(31,149),15,7
670 PUTSPRITE5,(30,164),1,8
680 PUTSPRITE6,(30,148),1,9
690 ONSPRITEGOSUB1310:SPRITEOFF
700 ONKEYGOSUB1410:KEY(1)ON
710 PUTSPRITE9,(202,157),1,11
720 JA=JA+1
730 IFJA=13ANDOP$="2"THEN1410
740 IFJA/2=INT(JA/2)THENJE=2ELSEJE=1
750 IFOP$<>"1"ANDJA/2=INT(JA/2)THENJI=2ELSEJI=1
760 PLAY"S1M500005L8CC.L16EL3D"
770 IF OP$<>"1"THENPA=76ELSEPA=85
780 LINE(222,PA)-(250,123),1,BF
790 IFOP$<>"1"THENPSET(220,77),1:PRINT#1,JE
800 PSET(219,92),1:PRINT#1,TI(JI)
810 PSET(219,107),1:PRINT#1,BU(JI)
820 CIRCLE(91,112),2,2:PAINT(91,112),2:CIRCLE(91,120),2,1:PAINT(91,120),1
830 IFINKEY$<>"9"THEN830
840 W=0
850 W=W-1E-04
860 IFINKEY$<>"0"THEN850
870 CIRCLE(91,120),2,9:PAINT(91,120),9:CIRCLE(91,112),2,1:PAINT(91,112),1
880 GOSUB1150
890 X=41:Y=132:T=1
900 X=X+1
910 IFY>141THENSPRITEONELSESPRITEOFF
920 T=T+W
930 Y=Y-T
940 PUTSPRITE0,(X,Y),1,10
950 IFY=<1ORX=>254THENGOSUB1320
960 IFY<164THEN900
970 PLAY"S1M500003L6C"
980 FORI=1TO500:NEXTI
990 PUTSPRITE0,(X,Y),0,10
1000 GOSUB1220
1010 GOTO710
1020 DATA192,240,255,224,168,128
1030 DATA28,84,126,252,252,124,56
1040 DATA3,15,31,63,63,127,127,247,231,247,247,119,55,7,7,0
1050 DATA192,224,224,224,224,224,224,224,224,224,224,224,224,224,224,0
```

```
1060 DATA240,240,240,240,240,240,240,240,240,240,240,240,240,240.240
1070 DATA126,126
1080 DATA192,255,255,255,255,224,224,224,224,224,224,224,224,224,224
1090 DATA192,224,240,248,252,254,238,230,226,224,224,224,224,224,224
1100 DATA192,224,224,240,240,248,248,252,252,248,224,224,224,224,224
1110 DATA32,32,32,32,32,32,32,32,32
1120 DATA1,7,7,15,15,63,127,255,255,255,127,127,61,1,3
1130 DATA0,128,224,224,240,240,252,254,255,255,255,254,254,188,128,192
1140 GOTO1140
1150 FORI=1TO3
1160 IFI=1THENF=38:G=142:ELSEIFI=2THENF=40:G=139:ELSEF=41:G=134
1170 PUTSPRITE10,(F,G),10,10
1180 PUTSPRITE3,(25,133),15,I
1190 FORT=1TO100:NEXT
1200 NEXTI
1210 RETURN
1220 FORI=3TO1STEP-1
1230 IFI=1THENF=38:G=142:ELSEIFI=2THENF=40:G=139:ELSEF=41:G=134
1240 PUTSPRITE10,(F,G),10,10
1250 PUTSPRITE3,(25,133),15,I
1260 FORT=1TO100:NEXT
1270 NEXTI
1280 TI(JI)=TI(JI)+1
1290 RETURN
1300 END
1310 SPRITEOFF:BU(JI)=BU(JI)+1:PUTSPRITE0,(100,10),7,10:PLAY"S1M1000007L3B":SPRI
TEON:GOSUB1220:GOTO710
1320 Y=Y+1
1330 PUTSPRITE0,(X,Y),1,10
1340 IFY>=164THEN1360
1350 GOTO1320
1360 PLAY"S1M500003L6C"
1370 GOSUB1220
1380 FORI=1TO400:NEXTI
1390 PUTSPRITE0,(X,Y),0,10
1400 GOTO710
1410 SPRITEOFF:SCREEN0:COLOR15,12
1420 PLAY"S1M500005L8CO4G.O5L16CL8D.L16CL8EC.L16EO4L8B.L16AL4G"
1430 IFOP$<>"1"THEN1490
1440 LOCATE2,2:PRINT"TIROS.........................";TI(1)
1450 LOCATE2,6:PRINT"BUENAS........................";BU(1)
1460 IFBU(1)=0THENRE=0ELSERE=(BU(1)*100)/TI(1)
1470 LOCATE2,10:PRINT"PORCENTAJE DE ACIERTO.........";(INT(RE*10))/10
1480 GOTO1580
1490 W1=2:W2=5:W3=7:W4=9
1500 FORI=1TO2
1510 LOCATE2,W1:PRINT"JUGADOR";I
1520 LOCATE5,W2:PRINT"TIROS..................";TI(I)
1530 LOCATE5,W3:PRINT"BUENAS.................";BU(I)
1540 IFBU(I)=0THENRE=0ELSERE=(BU(I)*100)/TI(I)
1550 LOCATE5,W4:PRINT"PORCENTAJE DE ACIERTO...";(INT(RE*10))/10
1560 W1=13:W2=16:W3=18:W4=20
1570 NEXT
1580 LOCATE0,22:PRINT"---------- PULSA UNA TECLA ----------"
1590 IFINKEY$=""THEN1590
1600 CLS
1610 PRINT:PRINT"  ¿Otra partida?    [1]"
1620 PRINT:PRINT"  ¿Listar juego?    [2]"
1630 PRINT:PRINT"  ¿Terminar juego? [3]"
1640 PRINT:INPUT"  OPCION (1,2,3)";R$
1650 IF R$="1"THEN RUN
1660 IF R$="2"THEN LIST
1670 IF R$="3"THEN NEW
```

Análisis del generador de sonidos

(segunda parte)

Continuando con lo expuesto en el número anterior, vamos a entrar en los detalles técnicos del chip. Asimismo, expondremos algunos ejemplos de los sonidos y ruidos más comunes, como colofón a este interesante tema.



El estándar MSX: especificaciones

Al leer el manual de referencia del MSX editado por Microsoft, encontramos varias referencias al PSG repartidas en diversos sitios. La primera de ellas aparece bajo la denominación general de sonido y dice:

a) El sonido habrá de tener ocho octavas y tres voces. Este es precisamente el alcance (8 octavas) y el número de canales (tres) que tiene el circuito integrado del que habla este artículo.

b) El Generador Programable de Sonidos habrá de ser compatible con el GI AY-3-8910. Esto quiere decir que habrá de ser dicho *chip*, o bien uno de otra marca que sea

exactamente igual en funcionamiento y patillaje.

c) El PSG funcionará con un reloj de 1.7897725 megahercios, que es un medio del reloj de la CPU. Esto quiere decir que el reloj que se inyecta en el AY-3-8910 es la mitad de rápido que el que inyecta en el microprocesador Z-80.

d) Deberá disponer de efectos sonoros. Esto es inmediato si se utiliza el circuito tantas veces mencionado, ya que disponemos de los efectos de volumen, envolvente y ruido.

e) Habrá una salida de sonido *software* de 1 bit por la puerta de entrad salida. Esto quiere decir que se usará uno de los bits de uno de los puertos del PSG para sacar señales de 1 bit («1» ó «0») para generar nuevos sonidos. Volveremos sobre ello cuando estudiemos las conexiones al exterior del PSG.



f) Si hibiese una salida al exterior de la señal generada por los tres canales del PSG, tendrá un nivel de –5dBm y se hará a través de un conector RCA de 2 *pines*. Esto es una referencia de cómo se ha de habilitar la posibilidad de grabar el sonido creado con un MSX, y es tal que permite ser grabado por un equipo de sonido convencional sin ningún problema, puesto que el conector es absolutamente estándar para estos equipos y el nivel es también el habitual.

La siguiente referencia al PSG que hallamos en dicho manual es la que se refiere al uso que se le dará, no específicamente relacionado con el sonido. En ella habla de la obliga-

toriedad de que exista un periférico de puertos de e/s, con uno o dos puertos habilitados, pudiendo dar cada uno de los puertos una corriente de 50 miliamperios.

Dicho periférico se realizará con un circuito integrado compatible con el GI AY-3-8910, disponiéndose en cada puerto de 4 bits de entrada, uno de salida y dos bidireccionales.

Esto se consigue por medio de una mezcla adecuada de los dos puertos de entrada/salida del PSG, como veremos en la siguiente sección.

Los bits de los puertos funcionarán con lógica activa en nivel alto (el «1» equivaldrá a una tensión de +5 voltios teóricos), siendo la distribución de niveles compatibles con TTL (lo que quiere decir que se interpreta como «1» cualquier tensión entre 2,5 y 5 voltios, y como «0» las tensiones entre 0 y 2 voltios).

Cada uno de estos puertos tendrá la forma física de un conector AMP de 9 *pines*, o un conector compatible. Este es el conector clásico para mando de juegos *(joysticks)* y, como vemos en la figura 9, ésta es la función principal a la que se va a dedicar. Hay cuatro bits que sirven para la entrada denominados *FWD* (adelante), *BACK* (atrás), *LET* (izquierda) y *RIGHT* (derecha) que se ponen a nivel bajo cuando se mueve un *joysticks* cualquiera en la dirección indicada. Los bits TRG1 y TRG2 (disparador 1 y 2) se corresponden con los disparadores que hay en un mando completo. Caso de tener el mando un solo disparador, únicamente nos será útil el bit denominado TRG1. Las salidas de + 5V y GND dan 5 y 0 voltios respectivamente, y sirven para alimentar al mando y lo que se conecte al puerto. Un mando de juegos habitual sólo usará la salida de GND para su funcionamiento, pero en algunos casos resulta útil tener + 5V, como por ejemplo en los mandos con disparo automático.

Para completar esta sección, mostramos en la figura 10 el esquema eléctrico habitual de un joystick y una posible realización del mismo.



BIT	7	6	5	4	3	2	1	0
FUNCION	Dirección del port: 0=entrada 1=salida		Suma de ruido 0=si			Tono 0=si		
CANAL	IOB	IOA	C	B	A	C	B	A

Figura 8. Registro general de control R7.

El estándar MSX: circuitos

En nuestro paseo por el estándar MSX, descubrimos que no todo lo

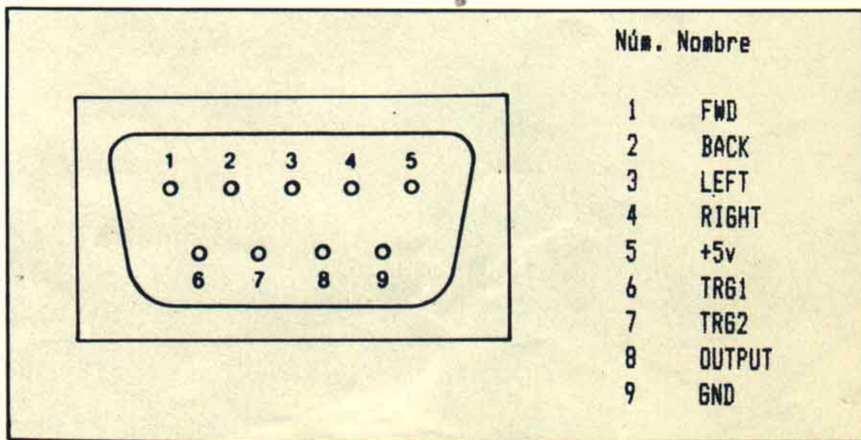


Figura 9. Conector del port de entrada/salida.





GANE
7.000 PTAS.
todos los meses
2E1554046
BANCO
DE
ESPAÑA
MIL
pesetas
1000
2E1554046
PARTICIPANDO
EN NUESTRO
CONCURSO
A partir del próximo número MSX premiará mensualmente los programas que hagan llegar los lectores.
Para participar en este concurso abierto, todo aficionado a los ordenadores con este estándar deberá hacer llegar a la redacción de la revista el listado, un cassette y un texto explicativo.
Entre todos los programas que recibamos cada mes, serán seleccionados para su publicación aquellos que reúnan los siguientes criterios:
— Originalidad de la aplicación.
— Simplicidad del método de programación.
La única condición para participar en el concurso será que los programas no hayan sido publicados previamente en ninguna revista.

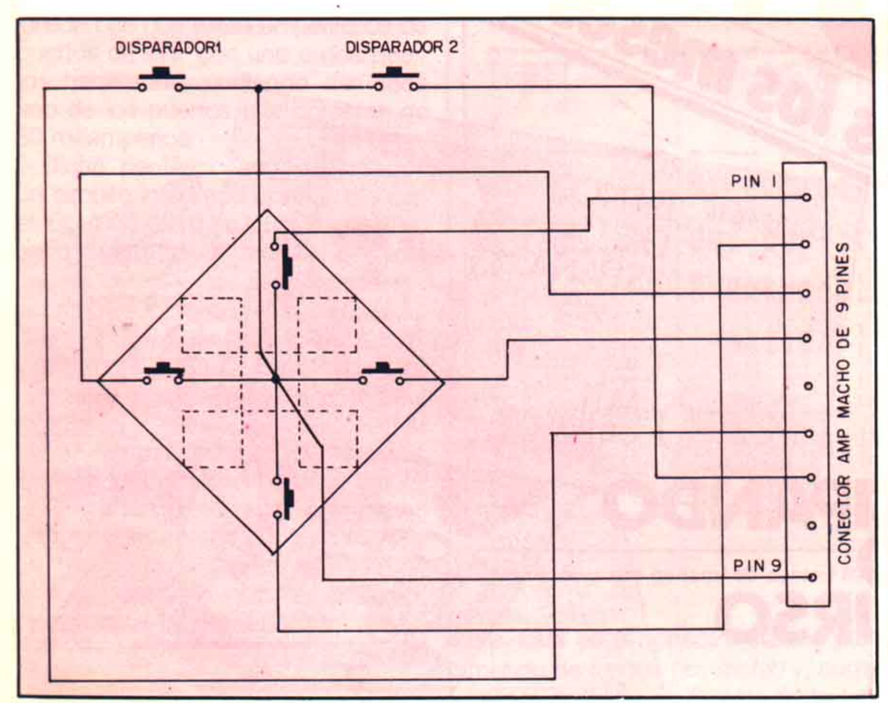


Figura 10. Mando de juegos para MSX.

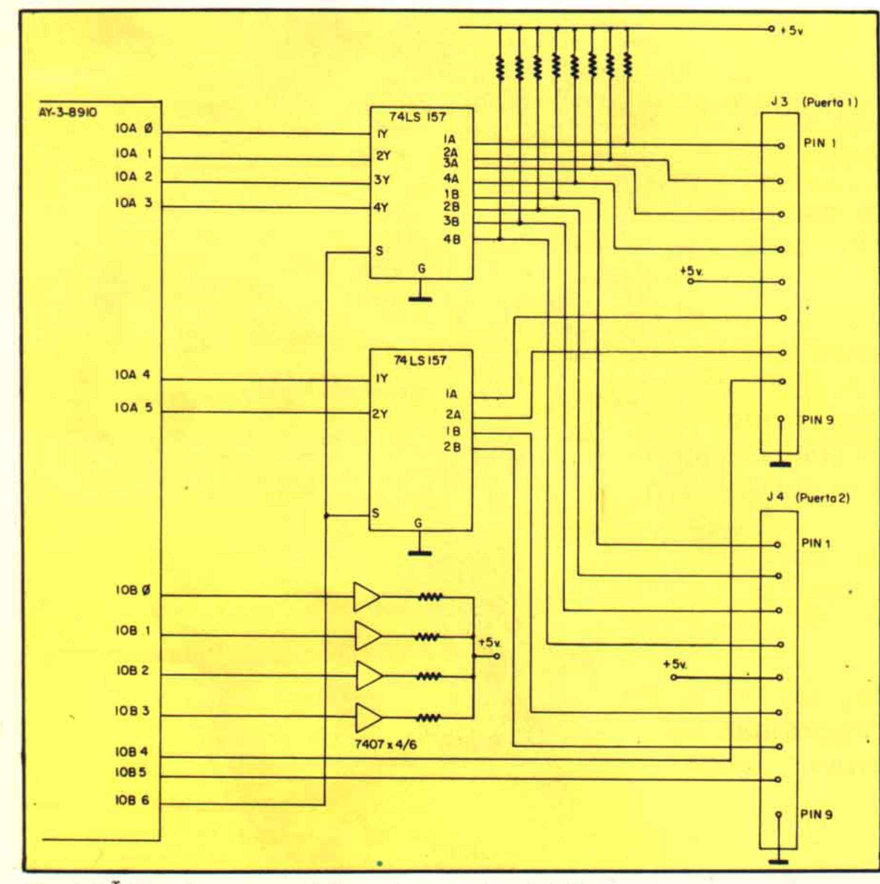


Figura 11. Circuito recomendado por los puertos de E/S.

que se dice sobre cómo construir un ordenador de la familia se hace en forma de descripciones, sino que también se SUGIEREN formas de conectar los circuitos entre sí. En relación con el PSG, aparece un solo esquema, y es el referido a los puertos de e/s (los puertos para conectar los mandos de juegos).

En la figura 11 podemos estudiar el esquema de dicha conexión. En ella aparecen varios circuitos integrados de los que conviene explicar su funcionamiento.

Los dos circuitos 74LS157 son de los llamados TTL por su forma de fabricación. Son *multipexores*, lo que quiere decir que, si ponemos la entrada marcada S a nivel lógico «0», se enviará a las salidas y entradas que existen en A, es decir, se conectarán 1A con 1Y, 2A con 2Y... Si ponemos un «1», se pasarán las entradas B a las salidas Y, pudiéndose leer el otro puerto. De este modo se podrá leer los valores de los dos mandos en los mismos bits del mismo puerto, el A (que funciona como sólo entrada) con sólo seleccionar mediante el bit 6 del puerto B, que funciona como «sólo salida».

Los bits 4 y 5 del puerto B van a dar al *pin* del conector denominado *OUTPUT* (salida), cuya utilidad puede ser muy variada (salida de sonido tipo Spectrum, por ejemplo) pero no evidente.

Los bits 0 a 3 del puerto B no se utilizan para nada, en principio, pero en vez de dejarlo al aire se los conecta a través de un *buffer*, el 7407, y una resistencia a la tensión de 5 voltios. Esto puede ser porque su utilidad actual es nula, pero podrán servir para algo en una futura ampliación del estándar MSX.

Los bits de 6 y 7 del puerto A, y el bit 7 del puerto B no aparecen en el dibujo, puesto que no se usan para nada que se refiera al mando de juegos. Sin embargo cuando hablamos en la sección anterior de bit de salida de sonido, es probable que se refieran a utilizar para ello el bit 7 del puerto B, o bien se refieran a los bits llamado *OUTPUT* (salida) de los conectores.

MSX:
LOS PERIFERICOS COMPATIBLES QUE CRECEN DIA A DIA.

Esta es la gran ventaja del sistema MSX: la total compatibilidad entre todos los periféricos de las distintas marcas con los diferentes ordenadores de este sistema que ya es standard mundial.
A todo esto, debemos añadir el respaldo y la garantía de SPECTRAVIDEO, creadores del sistema MSX.
He aquí una pequeña muestra:









MSX. DISC DRIVE

Unidad de disco, con controlador incluido, para diskettes de 5"1/4 de doble cara, doble densidad con capacidad de 500 kb (320 kb formateado). Preparado para su uso con el sistema MSX-DOS y CP/M. Utilizable con cualquier odenador del sistema MSX. (especialmente con el SVI-728).

MSX DATA CASSETTE

Grabador, reproductor a cassette, totalmente compatible para ser utilizado con cualquier ordenador MSX. Control Automático de Nivel (ALC), alta calidad de grabación. Parada automática. LED indicador. Bajo consumo.

MSX 80 COLUMNAS

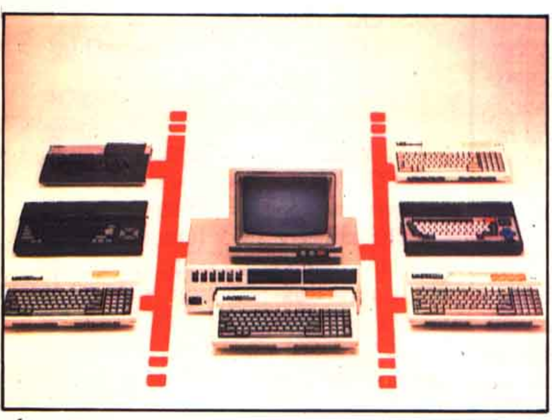
Cartucho de alta calidad que permite cambiar el Display de 40 a 80 columnas. Diseñado para ser utilizado especialmente con el ordenador SVI-728 u otros ordenadores del sistema MSX. Este cartucho, junto a una unidad de disco permite utilizar el sistema operativo CP/M.

MSX 64 K RAM

Diseñado para los ordenadores MSX dotados con una RAM inferior a 64 K, con lo que permite la utilización del sistema operativo CP/M. Imprescindible para la ejecución de programas que precisen gran cantidad de memoria.

MSX JOYSTICKS

De alta sensibilidad 100% compatibles con cualquier ordenador del sistema MSX. Fiabilidad total en 360°.
Modelo "QUICKSHOT I MSX"



MSX RED LOCAL DE COMUNICACIONES (LAN)

Permite trabajar hasta 32 ordenadores SVI-328, SVI-728 o cualquier otro standard MSX, que tenga, al menos, 64 K de memoria RAM.
Velocidad de transmisión 230 K/SEG.
Incluye un Winchester de 10 M. Conexión del sistema de gran sencillez y extremada facilidad de operación.
Solicite información en su Concesionario Autorizado Spectravideo.







DIREC.(hex.)	PERIFERICO		
00 - 3F	NO ESPECIFICADO	B4	RELOJ CALENDARIO
40 - 7F	RESERVADO	B5 - B7	NO ESPECIFICADO
80 - 87	RS-232C	B8 - BF	LAPIZ OPTICO
88 - 8F	NO ESPECIFICADO	C0 - CF	NO ESPECIFICADO
90 - 97	IMPRESORA	D0 - D7	CONTROLADOR DE DISCO
98 - 9F	VDP 9918A	D8 - DF	ROM DE CARACTERES KANJI
A0 - A7	PSG AY-3-8910	E0 - F6	NO ESPECIFICADO
A8 - AF	PPI 8255	F7	CONTROL DE AUDIO/VIDEO
B0 - B3	MEMORIA EXTERNA	F8 - FF	NO ESPECIFICADO

Figura 12. Mapa de direcciones de entrada y salida.

El estándar MSX: conexión con el Z-80

Para evitar que los programas de un ordenador MSX no puedan funcionar en otro, toda la entrada/salida del Z-80 ha de ir a los mismo sitios en todas las unidades que se saquen en el mercado. Microsoft ha definido totalmente el uso de las direcciones del espacio de entrada/salida del Z-80.

Para ello ha creado un «I/O ADDRES MAP» (Mapa de Direcciones de E/S), que podemos ver en la figura 12.

En él podemos ver que el PSG ocupa las direcciones &HA0 &HA8 (en decimal, 160 a 168). Estas direcciones no se utilizan completamente, sino que sólo se usan las direcciones &HA0 y &HA1.

Recordando lo que dijimos previamente sobre cómo escribir y leer el PSG, ya entonces advertimos que se escribía de dos modos y se leía de uno sólo. El primer modo de escribir, con BC1=1 para escribir el número de un registro al que se quiere acceder, y con BC1=0 para escribir al registro previamente seleccionado.

En un MSX, esto se consigue del siguiente modo:

a) Para escribir el número de un registro, escríbase dicho número en la dirección de e/s &HA0. Esto ya genera las señales adecuadas en BDIR (1), BC1 (1) y BC2 (1) para su correspondiente interpretación por el PSG.

b) Para escribir un dato en el registro previamente seleccionado, escribir dicho dato en la dirección e/s &HA1, lo que genera las señales BDIR=1, BC1=0 y BC2=1.

c) Para leer el valor de un registro previamente seleccionado, basta con leer el valor de la dirección &HA1 (al igual que para escribir), lo que genera BDIR=0, BC1=1, BC2=1.

El hecho de que leer de, y escribir a, los registros del PSG se realice en la misma dirección simplifica notablemente los contactos con el PSG, ya que cualquier operación que se quiera realizar con él se hará en dos etapas: una escritura en la dirección &HA0 seguida de una escritura o lectura en &HA1. En código máquina esto se realiza con una secuencia como la siguiente:

OPERACION DE LECTURA

El número de registro a leer está en el registro A.

```
OUT ($A0), A
IN A, ($A1)
```

el valor del registro se devuelve en a.

OPERACION DE ESCRITURA

El número de registro se encuentra en A
y el valor a escribir en él está en C

```
OUT ($A0), A
LD A,C
OUT ($A1), A
```

Así pues, lo que hace la instrucción *SOUND* del *BASIC* es escribir el número de registro en la dirección &HAS0 y el otro valor (entre o y 255) en &HA1. Para leer desde *BASIC* los valores de los registros, habremos de utilizar la instrucción *OUT* (similar al *out* del Z-80) y la función *PIN* (similar al *IN* del Z-80).

Algunos efectos curiosos

Vamos ahora a incluir algunos efectos curiosos que hemos extraído

del manual de referencia del AY-3-8910 editado por *General Instruments*. Los efectos no los ha creado esta redacción, pero sí la traducción a *BASIC* MSX con la sincronización necesaria. Se ha utilizado profusamente la instrucción *SOUND* A, B que, como ya hemos indicado, dá al registro A el valor B. Para que funcionen correctamente todos estos efectos, habrá que poner en el programa las siguientes líneas:

10000 FOR 1=0 TO 13: SOUND I, 0
NEXT I
10010 RETURN
cuyo fin s el de para cualquier sonido que hubiese anteriormente para que no interfiera.

1) Efectos de tono
Sirena Europea

```
10 GOSUB 10000
20 SOUND 0,254
30 SOUND 1,0
40 SOUND 7,62
50 SOUND 8,15
60 FOR I=1 TO 40:NEXT I
70 SOUND 0,86
80 SOUND 1,1
90 FOR I=1 TO 40:NEXT I:GOTO 20
```

2) Efecto de ruido
a) Disparo

```
10 GOSUB 10000
20 SOUND 6,15
30 SOUND 7,7
40 SOUND 8,16
50 SOUND 9,16
60 SOUND 10,16
70 SOUND 12,16
80 SOUND 13,0
90 FOR I=1 TO 500:NEXT I:STOP
```

b) Explosión

```
10 GOSUB 10000
20 SOUND 6,0
30 SOUND 7,7
40 SOUND 8,16
50 SOUND 9,16
60 SOUND 10,16
70 SOUND 12,56
80 SOUND 13,0
90 FOR I=1 TO 500:NEXT I:STOP
```

3) Efectos de secuencias
b) Láser

```
10 GOSUB 10000
20 SOUND 7,62
30 SOUND 8,15
40 FOR I=48 TO 112:SOUND 0,I:NEXT I
50 SOUND 8,0
60 GOSUB 10000:STOP
```

b) Bomba

```
10 GOSUB 10000
20 SOUND 7,62
30 SOUND 8,15
40 FOR I=48 TO 192:SOUND 0,I:NEXT I
50 GOSUB 10000
60 SOUND 6,15
70 SOUND 7,7
```

b) Coche de carreras

```
10 GOSUB 10000
20 SOUND 3,15
30 SOUND 7,62
40 SOUND 8,15
50 SOUND 9,10
60 SOUND 0,0:SOUND 1,11
70 FOR I=10 TO 4 STEP -1:SOUND 1,I:FOR J=255 TO 0 STEP
-1:SOUND 0,J:NEXT J:NEXT I
80 SOUND 0,0:SOUND 1,9
90 FOR I=8 TO 3 STEP -1:SOUND 1,I:FOR J=255 TO 0 STEP
-1:SOUND 0,J:NEXT J:NEXT I
100 SOUND 0,0:SOUND 1,6
110 FOR I=5 TO 1 STEP -1:SOUND 1,I:FOR J=255 TO 0 STEP
-1:SOUND 0,J:FOR K=0 TO 1:NEXT K:NEXT J:NEXT I
120 SOUND 8,0
130 SOUND 9,0
140 STOP
```

```
80 SOUND 8,16
90 SOUND 9,16
100 SOUND 10,16
110 SOUND 12,16
120 13,0
130 FOR I=1 TO 500:NEXT I:STOP
```

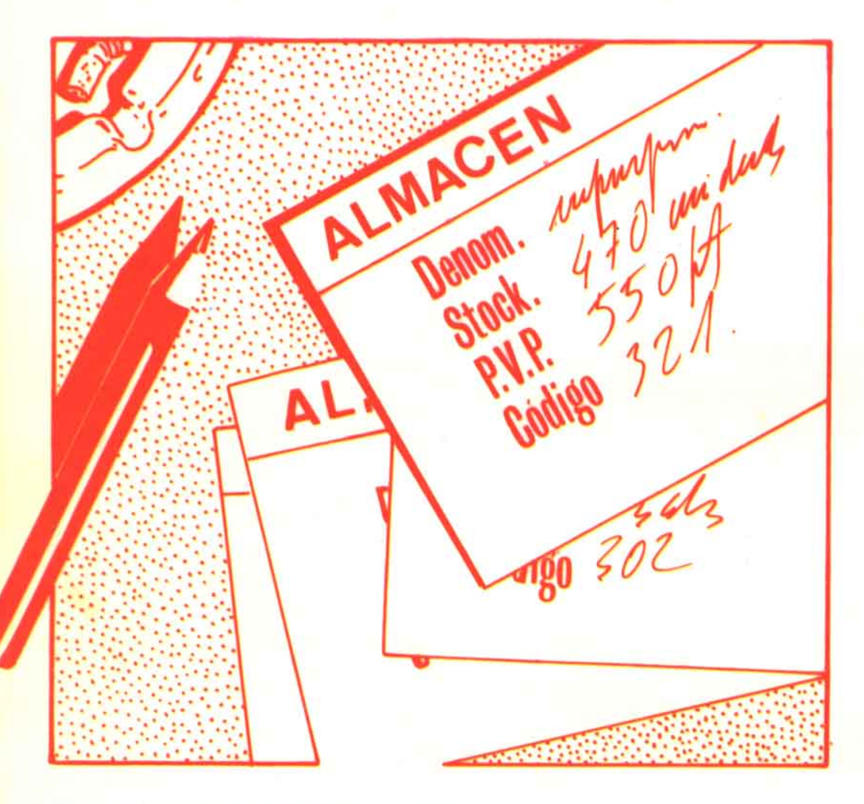
4) Efectos multicanal
a) Silbido

```
10 GOSUB 10000
20 SOUND 6,1
30 SOUND 7,46
40 SOUND 8,15
50 SOUND 9,9
60 FOR I=64 TO 32 STEP -1
70 SOUND 0,I:FOR J=0 TO 2:NEXT J:NEXT I
80 FOR I=1 TO 50:NEXT I
90 FOR I=64 TO 48 STEP -1
100 SOUND 0,I
110 FOR J=0 TO 5:NEXT J
120 NEXT I
130 FOR I=48 TO 104:SOUND 0,I:NEXT I
140 SOUND 8,0
150 SOUND 9,0
160 STOP
```

Este último efecto es el menos conseguido, ya que el *BASIC* no es suficientemente veloz, pero contiene todas las ideas necesarias para realizar el efecto en código máquina, aquellos que sean capaces. En cualquier caso, es interesante gastar algo de tiempo en teclearlo, ya que es una buena demostración de lo que puede el PSG AY-3-8910.

Juan Zarategui

Facturación



Para el que tenga una pequeña tienda o negocio, este programa no pasará inadvertido. La facturación es fundamental a la hora de realizar los cálculos necesarios para obtener el beneficio o pérdida del comerciante. En este caso, el programa ayuda a llevar un control de los clientes.

Cuando el cliente compra, introduciremos todos los datos necesarios tales como el número, nombre, dirección, población, así como el descuento y el I. T. E.

Con estos datos, pasaremos a introducir el artículo, su descripción, unidades, el precio/unidad, obteniendo como resultado el total por artículo.

A continuación, pasaremos a la segunda pantalla, donde se nos muestra el nombre y apellido del cliente y todos los totales, el Total Bruto, Líquido, Descuento, etc. Seguidamente se nos preguntará CARGO o ABONO a la operación de dicho cliente. Después de haber contestado, proseguirá con portes, embalajes, etc., que irán acumulando o descontando el Total Líquido.

José Manuel Vera Vilchés
Córdoba

```
1 REM ********************************
2 REM * (C) JOSE MANUEL VERA VILCHES *
3 REM *              PARA            *
4 REM *          MSX MAGAZINE        *
5 REM ********************************
10 CLS
20 SCREEN 0:COLOR 15,1
30 WIDTH 40: KEY OFF
40 CLEAR 6000
50 PRINT"┌─────────────────────────────────────┐"
60 FOR G=1 TO 19
70 LOCATE 0,G:PRINT"|";SPACE$(37);"|"
80 NEXT G
90 PRINT"└─────────────────────────────────────┘"
100 LOCATE 0,0:PRINT"┌─────────┐"
110 LOCATE 0,1:PRINT"|N.DE CLTE|"
120 LOCATE 0,2:PRINT"└─────────┘"
130 LOCATE 0,2:PRINT"┌──────┐"
140 LOCATE 0,3:PRINT"|NOMBRE|"
150 LOCATE 0,4:PRINT"└──────┘"
160 LOCATE 0,4:PRINT"┌─────────┐"
170 LOCATE 0,5:PRINT"|DIRECCION|"
180 LOCATE 0,6:PRINT"└─────────┘"
190 LOCATE 0,6:PRINT"┌─────────┐"
200 LOCATE 0,7:PRINT"|POBLACION|"
```

Programas Sony para ordenadores MSX

A la orden.


Monkey Academy


Países del Mundo-1


Países del Mundo-2


Computador Adivino


Computer Billiards


The Snowman


Cubit


Character Collection


Stop the express (Para el Tren)


Hustler (Billar Americano)


Data cartridge


Quinielas y Reducciones


Home Writer


Sparkie


Aprendiendo Inglés-1


Binary Land


Creative Greetings


Aprendiendo Inglés-2


Antartic Adventure


Mastermind


Contabilidad Personal


Athletic Land


E.I.


Ficheros


El Ahorcado


Dorodon


La Pulga


Cosmos


Control de Stocks


Battle Cross


Mouser


Crazy Train


Ali baba


Juno First


Car Jamboree


Tutor


Track and Field-1 (olimpiadas)


Blackjack


Track and Field-2 (olimpiadas)


Driller Tanks (Tanque Destructor)


Sonygraph


Ninja (El Samurai)



Les Flics



Y muchos más títulos

Ordenador Doméstico

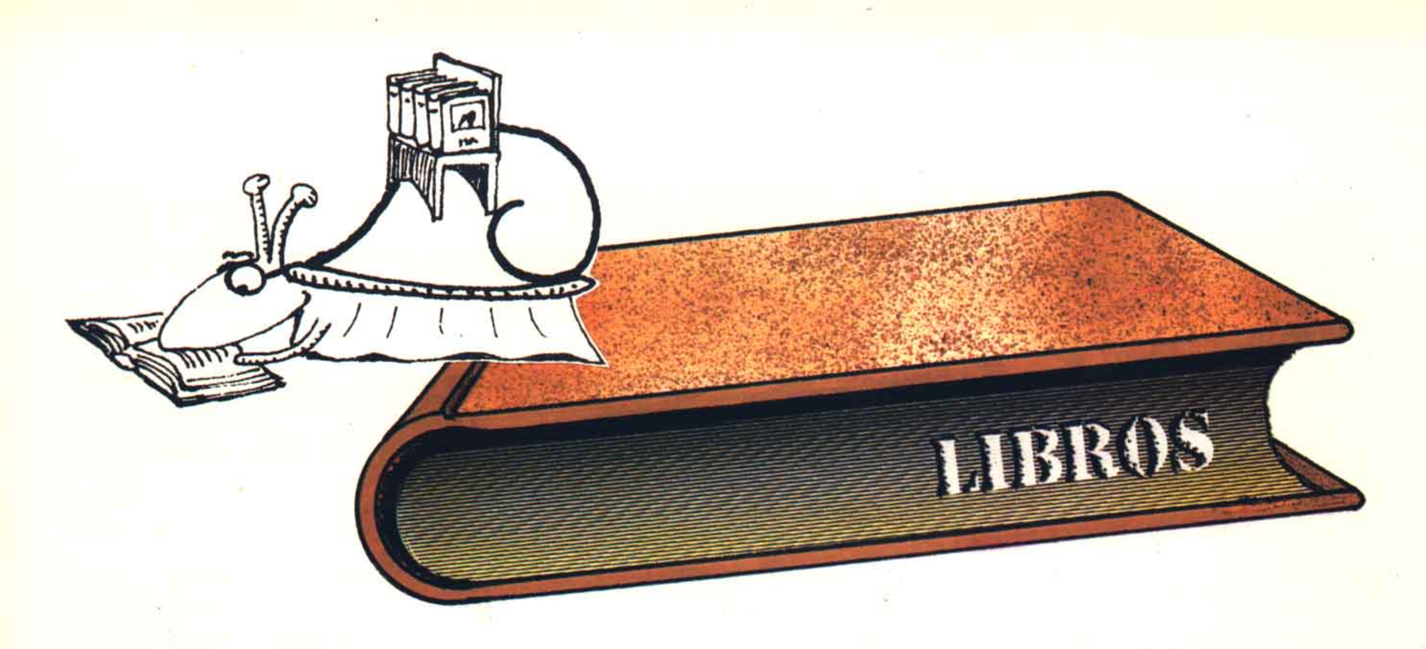
HIT BIT
SONY

Para lo que guste ordenar.



```
210 LOCATE 0,8:PRINT"└──────────┘"
220 LOCATE 24,0:PRINT"┌──────────┐"
230 LOCATE 24,1:PRINT"|DESCUENTO|"
240 LOCATE 24,2:PRINT"└──────────┘"
250 LOCATE 26,2:PRINT"┌──────┐"
260 LOCATE 26,3:PRINT"|I.T.E.|"
270 LOCATE 26,4:PRINT"└──────┘"
280 FOR I=0 TO 1  STEP 1
290 LOCATE 1,8+I:PRINT"────────────────────────────────────"
300 NEXT I
310 FOR J=1 TO 37
320 LOCATE J,8:PRINT"▄"
330 LOCATE J,9:PRINT"▀"
340 NEXT J
350 LOCATE 1,10:PRINT"ART.|"
360 FOR T=11 TO 19
370 LOCATE 1,T:PRINT"    |"
380 NEXT T
390 LOCATE 6,10:PRINT"DESCRIPCION|"
400 FOR T=11 TO 19
410 LOCATE 6,T:PRINT"           |"
420 NEXT T
430 LOCATE 18,10:PRINT"UNID|"
440 FOR T=11 TO 19
450 LOCATE 18,T:PRINT"    |"
460 NEXT T
470 LOCATE 23,10:PRINT"P/UND|"
480 FOR T=11 TO 19
490 LOCATE 23,T:PRINT"     |"
500 NEXT T
510 LOCATE 29,10:PRINT"  TOTAL "
520 LOCATE 29,11:PRINT"  ----- "
530 LOCATE 11,1: LINE INPUT"";N$
540 LOCATE 8,3:LINE INPUT"";NO$
550 LOCATE 11,5:LINE INPUT"";D$
560 LOCATE 11,7:LINE INPUT"";P$
570 LOCATE 35,1:LINE INPUT"";E$
580 LOCATE 34,3:LINE INPUT"";I$
590 LOCATE 2,21:PRINT"CORRECTO (S/N)"
600 X$=INPUT$(1)
610 IF X$="S" OR X$="s" THEN LOCATE 2,21:PRINT SPACE$(14):GOTO 630
620 IF X$="N" OR X$="n" THEN LOCATE 11,1:PRINT SPACE$(13):LOCATE 8,3:PRINT SPACE
$(18):LOCATE 11,5:PRINT SPACE$(27):LOCATE 11,7:PRINT SPACE$(27):LOCATE 35,1:PRIN
T SPACE$(3):LOCATE 34,3:PRINT SPACE$(4):LOCATE 2,21:PRINT SPACE$(14):GOTO 530:EL
SE 600
630 F=12
640 FOR I=1 TO 200
650 LOCATE 1,F:LINE INPUT"";AR$ '--ARTICULO
660 AR#=VAL(AR$)
670  LOCATE 1,F:PRINT USING"####";AR#
680 LOCATE 7,F:LINE INPUT"";DES$ '--DESCRIPCION
690 LOCATE 19,F:LINE INPUT"";UN$ '--UNIDADES
700 UN#=VAL(UN$)
710 LOCATE 19,F:PRINT USING"###";UN#
720 LOCATE 24,F:LINE INPUT"";PU$ '--PRECIO POR UNIDAD
730 PU#=VAL(PU$)
740  LOCATE 24,F:PRINT USING"###";PU#
750 T#=UN#*PU#:IF T#=0 THEN 870
760 LOCATE 30,F:PRINT USING"######";T#  '--TOTAL
770 LOCATE 2,21:PRINT "CORRECTO (S/N)"
780 X$=INPUT$(1)
790 IF X$="N" OR X$="n" THEN LOCATE 2.21:PRINT SPACE$(14):LOCATE 1,F:PRINT SPACE
```

```
$(4):LOCATE 7,F:PRINT SPACE$(10):LOCATE 19,F:PRINT SPACE$(3):LOCATE 24,F:PRINT S
PACE$(3):LOCATE 30,F:PRINT SPACE$(6): GOTO 650
800 IF X$="S" OR X$="s" THEN LOCATE 2,21:PRINT SPACE$(14):F=F+1:ELSE 780
810 IF F=19 THEN F=12:GOTO 840
820 MO#=MO#+T#
830 NEXT I
840 FOR K=12 TO 19
850 LOCATE 1,F:PRINT SPACE$(4):LOCATE 7,F:PRINT SPACE$(10):LOCATE 19,F:PRINT SPA
CE$(3):LOCATE 24,F:PRINT SPACE$(3):LOCATE 30,F:PRINT SPACE$(6):F=F+1
860 NEXT K:F=F+1:F=12:GOTO 830
870 CLS:SCREEN 0
880 PRINT "┌─────────────────────────────────────┐"
890 FOR T=1 TO 15
900 PRINT "|";SPACE$(37);"|"
910 NEXT T
920 PRINT "└─────────────────────────────────────┘"
930 LOCATE 10,0:PRINT "TOTALES DE FACTURA"
940 LOCATE 2,3:PRINT"CLIENTE NRO...:";N$:LOCATE 25,3:PRINT SPACE$(20):LOCATE 38.
3:PRINT "|";SPACE$(1);"|"
950 LOCATE 2,4:PRINT"NOMBRE CLTE...:";NO$:LOCATE 35,4:PRINT SPACE$(20):LOCATE 38
,4:PRINT "|";SPACE$(1);"|"
960 LOCATE 2,5:PRINT"DIRECCION CLTE:";D$:LOCATE 38,5:PRINT SPACE$(10):LOCATE 38,
5:PRINT "|";SPACE$(1);"|"
970 LOCATE 2,6:PRINT"POBLACION CLTE:";P$:LOCATE 35,6:PRINT SPACE$(10):LOCATE 38,
6:PRINT"|";SPACE$(1);"|"
980 LOCATE 2,7:PRINT"I.T.E.........:";I$:LOCATE 20,7:PRINT SPACE$(2)
990 LOCATE 2,8:PRINT"DESCUENTO.....:";E$:LOCATE 20,8:PRINT SPACE$(2)
1000 FOR I=1 TO 37
1010 LOCATE I,10:PRINT"▀"
1020 NEXT I
1030 E=VAL(E$)
1040 TA#=MO#*E/100
1050 LOCATE 2,12:PRINT"TOTAL BRUTO..:";MO#
1060 LOCATE 2,13:PRINT"TOTAL DESCUENTO:";TA#
1070 I=VAL(I$):LO#=MO#*I/100
1080 LOCATE 2,14:PRINT"TOTAL I.T.E....:";LO#
1090 TL#=MO#-TA#+LO#
1100 LOCATE 2,15:PRINT"TOTAL LIQUIDO..:";TL#
1110 FOR I=1 TO 37
1120 LOCATE I,17:PRINT"▀"
1130 NEXT I
1140 LOCATE 0,20:PRINT SPACE$(20):LOCATE 10,19:PRINT"CARGO O ABONO (S/N)"
1150 X$=INPUT$(1)
1160 IF X$="S" OR X$="s" THEN 1180
1170 IF X$="N" OR X$="n" THEN 1220 ELSE 1150
1180 LOCATE 10,19:PRINT"  (C)ARGO-(A)BONO  "
1190 X$=INPUT$(1)
1200 IF X$="C" OR X$="c" THEN INPUT"CANTIDAD:";C:TL#=TL#+C:LOCATE 18,15:PRINT TL
#:GOTO 1140
1210 IF X$="A" OR X$="a" THEN  INPUT"CANTIDAD:";A:TL#=TL#-A:LOCATE 18,15:PRINT T
L#:GOTO 1140:ELSE 1190
1220 LOCATE 0,20:PRINT SPACE$(20):LOCATE 10,19:PRINT "     PORTES(S/N)      "
1230 X$=INPUT$(1)
1240 IF X$="S" OR X$="s" THEN 1260
1250 IF X$="N" OR X$="n" THEN 1270 ELSE 1230
1260 INPUT "PORTES...:";P:TL#=TL#+P:LOCATE 18,15:PRINT TL#:GOTO 1220
1270 LOCATE 0,20:PRINT SPACE$(20):LOCATE 10,19:PRINT "  EMBALAJES(S/N)  "
1280 X$=INPUT$(1)
1290 IF X$="S" OR X$="s" THEN 1310
1300 IF X$="N" OR X$="n" THEN 1320 ELSE 1280
1310 INPUT"EMBALAJES...:";E:TL#=TL#+E:LOCATE 18,15::PRINT TL#:GOTO 1270
1320 LOCATE 8,21:PRINT "CONCLUSION DEL TRABAJO"
```



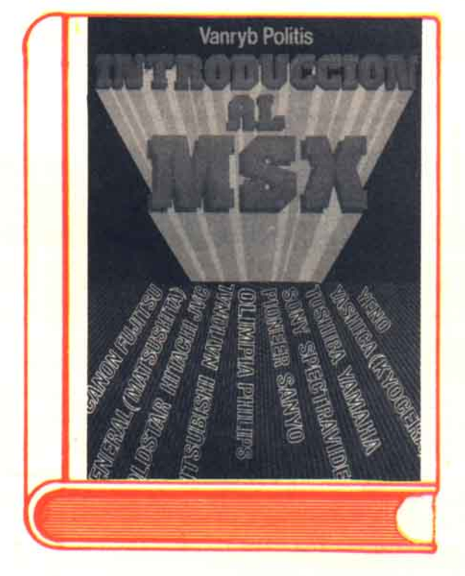
Título: Introducción al MSX
Autor: Vanryb Politis
Editorial: Noray
Páginas: 192
Precio: 1.250 Ptas.

Como su nombre indica, se trata de un libro que ayudará al principiante a introducirse en las características del BASIC del estándar MSX.
A lo largo de 192 páginas, se explican todas las características del lenguaje y sus aplicaciones.

El profano agradecerá dos buenas cualidades del libro. En primer lugar, los innumerables ejemplos que se exponen en los diversos capítulos ayudan de una manera sencilla a asentar los conceptos que se explican en cada caso y, en segundo lugar, el libro está escrito en un lenguaje coloquial, sin tecnicismos que compliquen los temas.

El libro se divide en dos partes. La primera, que consta de dos capítulos, es una introducción al MSX, donde se explican temas que van desde la puesta en marcha del ordenador hasta la estructura de un programa, analizando las variables, bifurcaciones, etc.

Todos los conceptos de esta primera parte son muy simples, y están enfocados fundamentalmente para aquellos que no han tenido contacto con lenguajes de programación.



La segunda, algo más compleja, trata del BASIC del MSX en todos sus aspectos, desde las especificaciones del lenguaje hasta las características del sonido, pasando por un capítulo muy atractivo como es la gestión de periféricos, dentro del cual se comentan las posibilidades del MSX-DOS, algo a tener en cuenta si en un futuro decide ampliar su equipo con una unidad de disco, muy útil en todos los aspectos. Hay que destacar que las instrucciones están divididas en familias, lo que permitirá encontrar cualquier comando concreto.

Al final del libro hay tres apéndices en los que encontaremos el juego de caracteres, los mensajes de error y las características técnicas del MSX.

El libro está bien estructurado, aunque la presentación podía haberse mejorado en parte, quizás ilustrando los capítulos con dibujos. De cualquier manera, esto no plantea el menor problema, pues, como ya dijimos anteriormente, los innumerables ejemplos que incluye el libro ayudan a comprender, incluso mejor que con figuras, cualquier tema que se esté analizando.

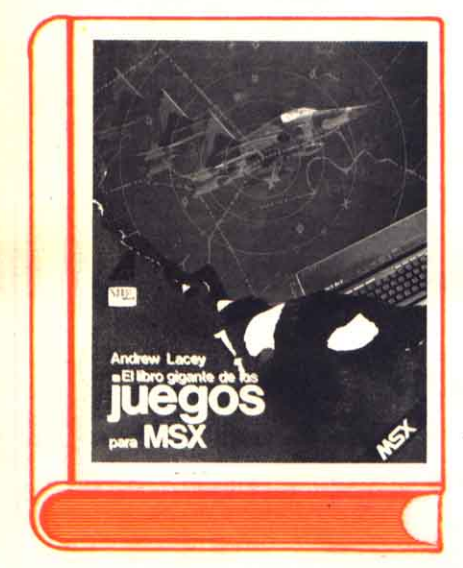
Título: El libro de los juegos para MSX
Autor: Andrew Lacey
Editorial: Anaya Multimedia
Páginas: 302
Precio:

No podía faltar un libro de juego dentro de la diversidad de ofertas con que nos encontramos. En este caso, Anaya ha reunido más de 25 programas y los ha expuesto de forma que sean fáciles de comprender.

En las primeras páginas nos encontramos con una pequeña introducción que aclara la simbología que

se utiliza en las explicaciones de cada programa, ya que éstos vienen descritos detalladamente, con separación de bloques de instrucciones, ideas para mejorar el programa, explicación de las variables, etc., pero lo más significtivo del libro es que al principio y antes de que lector continúe su andadura por las páginas en cuestión, se comenta cuidadosamente un programa corto que permitirá detectar cualquier error que exista en la transcripción del listado. Este programa indica la dirección del comienzo del BASIC y está preparado para realizar las comprobaciones del resto de los programas del libro, con lo que el lector se ahorrará muchos problemas.

Ahora bien, para que este programa funcione correctamente conviene dedicarle especial atención, ya que los minutos que se pierden al principio, a la larga se ganan con creces, pues muchas veces nos veremos en la situación de creer que está correcto el programa cuando en realidad le falta un punto, una variable, o cualquier elemento en alguna instrucción.



Es interesante destacar que existen diversos tipos de programas a lo largo del libro, ya que no sólo contiene juegos, sino también programas educativos, interactivos, memorísticos, etc...

Las buenas ilustraciones que se muestran en los distintos programas, sirven de guía al usuario atrevido que quiera alterar cualquier *sprite* incluído en los programas.

Título: Despega con tu MSX
Autor: Manuel Medel
Editorial: Peopleware S.A.
Páginas: 241
Precio: 2.300 Ptas.

Aprender jugando ha sido siempre un tópico difícil de lograr por aquellos que se dedican a la enseñanza, pero está claro que cualquier cosa que se realice dentro de unas condiciones óptimas (buen ambiente, gusto a lo que se hace, etc...). es mucho más fácil de asimilar que hacer el mismo trabajo en condiciones totalmente adversas. Por lo tanto, aprender jugando o jugando a aprender, no es una idea nueva.

Sin embargo, plasmar el concepto de MSX, su lenguaje y sus aplicaciones en dos libros (de los cuales comentamos este mes el primero de ellos), sí es una buena idea.

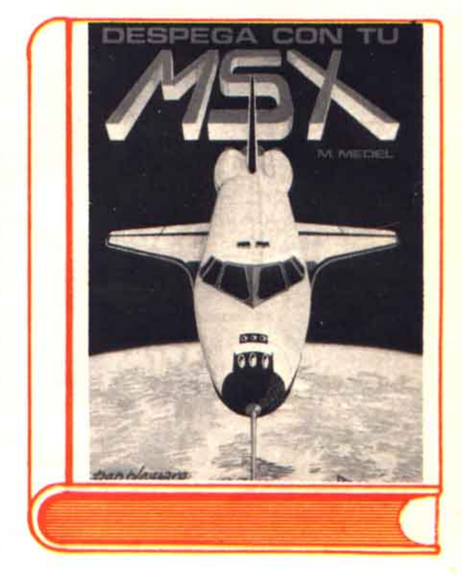
A lo largo de 250 páginas, el lector se verá sumergido, no sólo en un libro que explica el BASIC del MSX y todos sus conceptos, instrucciones y hasta sus aplicaciones, sino en una aventura que verá crecer y desarrollarse delante de uno mismo.

Efectivamente, con el libro no se pretende dar a conocer a fondo las posibilidades del MSX, sino que uno se diverte viendo y comprobando como se va a desarrollar un programa muy particular: un auténtico simulador de vuelo.

Indudablemente, un programa de estas características no es nada nuevo, pero sí lo es cuando se trata de la nave que es réplica del Columbia, (el famoso transbordador americano) y se encuentre dentro de ella volando en busca de satélites averiados.

El lector comprobará cómo los conocimientos adquiridos capítulo a capítulo, no caen en saco roto, sino que se van interrelacionando entre sí. Cuando el lector pasa de un tema a otro, no se encuentra descolgado ni perdido, todo se va acumulando para lograr un fin: que la nave despegue y cumpla su misión.

El libro está bien estructurado y preparado. Se inicia, como en todos los casos, con una breve explicación del estándar MSX y con unas ideas de lo que es la programación estructurada.

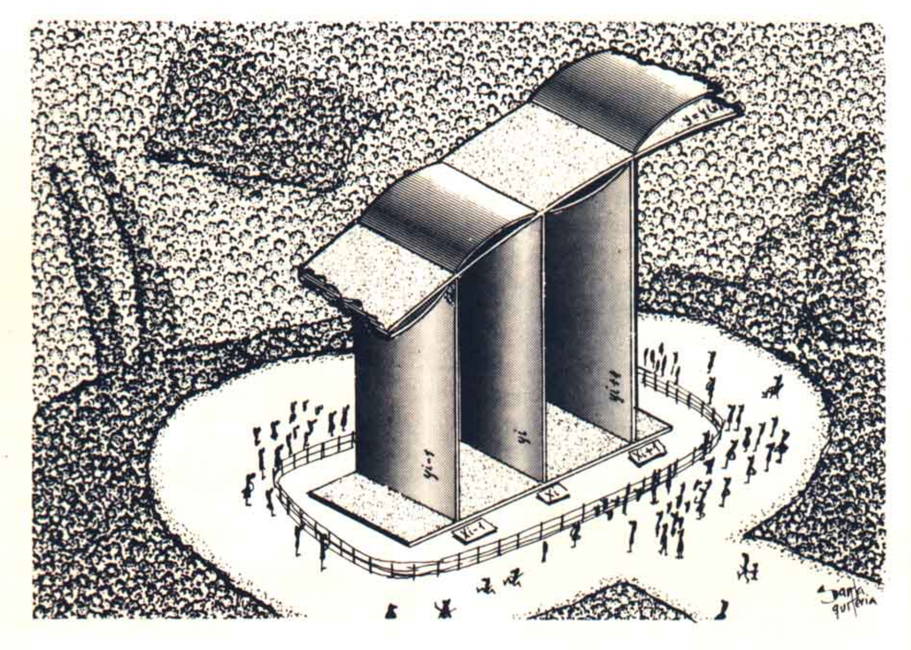


A continuación se comienzan a explicar las instrucciones y características del MSX, de forma que, con el libro y el ordenador conectado, se vayan asimilando los conocimientos que se han adquirido a lo largo de la lectura.

El programa a realizar se va explicando paso a paso mediante bloques, lo que sin duda beneficiará al lector, ya que de una forma sencilla se le va introduciendo en la técnica de la programación estructurada.

Por último hemos de hacer mención de la excelente presentación gráfica de las instrucciones del libro, que van mostrando a lo largo del desarrollo del programa lo que se está realizando en cada momento, y cómo debe quedar en la pantalla después de realizar un bloque determinado.

Método de Simpson



Este programa es de gran utilidad en la investigación, ya que gran parte de las funciones obtenidas experimentalmente no tienen una ecuación matemática asignable.

La ecuación que se aplica en el programa es:

$$\text{Area} = \frac{T - R}{3 * n} * (y0 + yn + 4\ \text{y impares} + 2\ \text{y pares})$$

siendo T y R los valores límite superior e inferior respectivamente, de la zona para las x. N es el número de intervalos dentro de la citada zona y habrá de ser par.

Si la secuencia de valores tomados para la función es: y0, y1, y2, y3,..., yn, se consideran impares las funciones de subíndice impar excluída la última y pares las de subíndice impar excluída la primera.

La estructura del programa es sencilla. Hasta la línea 55 se introducen los valores de T, R y N. Los valores de la función se introducen en bloque en el bucle de las líneas 80 - 90, obteniéndo los sumarios parciales en los dos siguientes bucles 95 - 120 y 130 - 150.

El resto del programa controla la salida de los resultados y la impresión de éstos.

José Fco. Cornejo Montalvo
Valencia

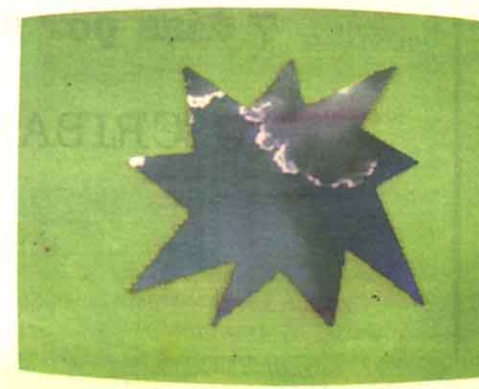
```
10 REM"INTEGS"
20 REM JOSE FRCO. CORNEJO
25 REM METODO DE SIMPSON
30 CLS
32 INPUT"LIMITE INFERIOR DE LAS X";R
34 PRINT:PRINT
40 INPUT"LIMITE SUPERIOR DE LAS X";T
45 IF R>T GOTO 32
47 PRINT:PRINT
50 INPUT"NUMERO DE INTERVALOS IGUALES(ELIGE NUMERO PAR)";N
55 IF INT(N/2)*2<>N GOTO 50
57 PRINT:PRINT
```

```
60 PRINT"LA SECUENCIA DE FUNCIONES ES:yo,y1,y2,y3,...,yn"
65 PRINT:PRINT
70 DIM A(N)
75 PRINT"INTRODUCE TODOS LOS VALORES DE LA FUNCION SEGUN LA SECUENCIA ESTABLECID
A"
80 FOR I=0 TO N
85 INPUT A(I)
90 NEXT I
92 REM SUMATORIO DE VALORES IMPARES DE LA FUNCION
95 FOR I=1 TO N-1 STEP 2
100 H=H+A(I)
120 NEXT I
125 REM SUMATORIO DE VALORES PARES DE LA FUNCION
130 FOR I=2 TO N-2 STEP 2
140 S=S+A(I)
150 NEXT I
160 M=(T-R)*(A(0)+A(N)+4*H+2*S)/(3*N)
165 PRINT:PRINT
170 PRINT" EL VALOR DE LA INTEGRAL ES";M
```

VIDEO ART

Aunque pudiera parecer que existen muy pocas diferencias entre los modos de pantalla *SCREEN* 0 y *SCREEN* 1, lo cierto es que sus semejanzas son verdaderamente escasas. Sólo tienen en común que son modos de pantalla para texto y que utilizan el mismo juego de caracteres. Por lo demás, todo son distinciones: el modo *SCREEN* 1 trata los colores de forma muy diferente, acepta figuras móviles, amplía el tamaño de los caracteres, disminuye el número de columnas, etcétera...

Las 192 filas × 256 columnas de *píxels* que tiene la pantalla en todos los modos *SCREEN*, son distribuidos por el modo *SCREEN* 1 en cuadros de 8 filas × 8 columnas de *píxels*, a diferencia de las 6 columnas de los cuadros del modo *SCREEN* 0. Al ser



La memoria de video: caracteres en color (II)

El mes pasado vimos las tablas que componen el sistema de almacenamiento en el modo SCREEN 0 y sus aplicaciones. Este mes veremos las tablas y características correspondientes al modo SCREEN 1.

de este tamaño, queda establecido en pantalla un plano coordenado de 24 filas × 32 columnas de cuadros.

La tabla 5: ¿Qué hay en la pantalla?

El funcionamiento de la tabla número 5 es muy parecido al de la tabla 0, que exponíamos en el número anterior. Su longitud es de 768 octetos, uno por cada uno de los cuadros de la pantalla (24 × 32 = 768). Como sucede en la tabla 0, los octetos de la tabla 5 tienen la función de almacenar el número del carácter que se halla en su cuadro correspondiente.

En nuestros ordenadores MSX la tabla 5 comienza en el octeto 6144 y termina en el 6911, ambos inclusive. Estos 768 octetos se dividen en 24 grupos de 32 octetos, un grupo por cada línea de pantalla, de tal forma que el octeto 6144 corresponde al cuadro (0,0), el 6145 al cuadro (1,0), y así hasta el octeto 6175, que corresponde al cuadro (31,0). El octeto 6176 corresponde al cuadro (0,1), y en esta doble sucesión podríamos seguir hasta el octeto 6911, correspondiente al (31,23), que es el cuadro de la esquina inferior derecha.

Para hallar el octeto correspondiente a un cuadro en función de su número de línea y de columna, es necesario efectuar la siguiente operación: primer octeto de la tabla + n.° de línea × 32 + n.° de columna.

Al estar dispuestos de este modo los cuadros en el modo *SCREEN* 1, si se quiere disponer de la totalidad de la pantalla es necesario efectuar un *WIDTH* 32, y no un *WIDTH* 40, como en el modo *SCREEN* 0, y no se pueden utilizar comandos *WIDTH* que ordenen un mayor número de columnas.

Hasta aquí un breve resumen de los aspectos de la tabla 5 que no coinciden con la tabla 0. Por lo demás, funciona igual, y sus aplicaciones prácticas con la tabla 0. Por lo demás, funciona igual, y sus aplicaciones prácticas son las mismas. Para ampliar información sobre el mismo e investigar sobre sus múltiples aplicaciones, sería necesario consultar el número anterior.

Las tablas 6 y 7: el tratamiento del color

Los ordenadores pertenecientes al estándar MSX funcionan principalmente en base a tres colores: el color de la tinta, el del fondo y el del



borde. El color del borde es el color de la zona de la pantalla que rodea al rectángulo de 256 × 192 *píxels* de que dispone tu ordenador. En esta zona no es posible pintar, ni escribir, ni dibujar: sólo es posible poner TODA LA ZONA DE BORDE de un color concreto (por ejemplo, el verde) de esta manera:
COLOR ,,2

Los otros dos colores, el color de tinta y el color de fondo, son muy importantes en la salida a pantalla, pues gracias a ellos el ordenador se puede comunicar con el programador. Si logramos distinguir una letra (la «A» mayúscula, por ejemplo) es gracias a que los *píxels* que perfilan «A» están pintados del color de la tinta y el resto de los *píxels* de este cuadro se hallan del color de fondo. El efecto que posee el modo *SCREEN* 0 en este aspecto es que sólo puede admitir un color de fondo y un color de tinta en toda la pantalla, mientras que en el modo *SCREEN* 1 es posible situar todos los colores gracias a la tabla 6. Pero para llegar a entender la tabla 6, es necesario aprender antes el funcionamiento de la tabla 7.

La tabla 7 almacena los datos exactamente igual que la tabla 2 en el modo *SCREEN* 0. Aquella ocupa, como ésta, 2048 octetos, destinados a mantener en memoria los patrones de los 256 caracteres, conteniendo cada patrón un total de ocho octetos. Cada octeto representa una fila de *píxels* del carácter. Si transformamos al sistema binario el número que contiene el octeto, hallaremos que tiene un máximo de ocho cifras binarias, representantes de cada uno de los ocho *píxels* de una fila.

Si la cifra a la que corresponde el *píxel* es «1», el *píxel* se pintará del color de la tinta, y si es el «0» se pintará del color de fondo.

Hasta aquí, la tabla 7 funciona igual que la tabla 2, pero hay diferencias que conviene destacar. Por ejemplo, la tabla 2 comienza en el octeto 2048 en *SCREEN* 0, pero la tabla 7 comienza en el octeto 0 en

Las 192 filas x 256 columnas de píxels son tratados de distintas formas, según el modo de pantalla.

SCREEN 1, terminando lógicamente en el octeto 2047 del sistema de *screen* 1. Otra diferencia es que los patrones de los caracteres pueden ser elaborados sabiendo que las dos columnas de *píxels* de la derecha SI VAN A SALIR EN PANTALLA, gracias a que en el modo *SCREEN* 1 los cuadros tienen ocho columnas de *píxels* y no seis.

La fórmula para hallar el primer octeto de un carácter es la misma que la de la tabla 2:

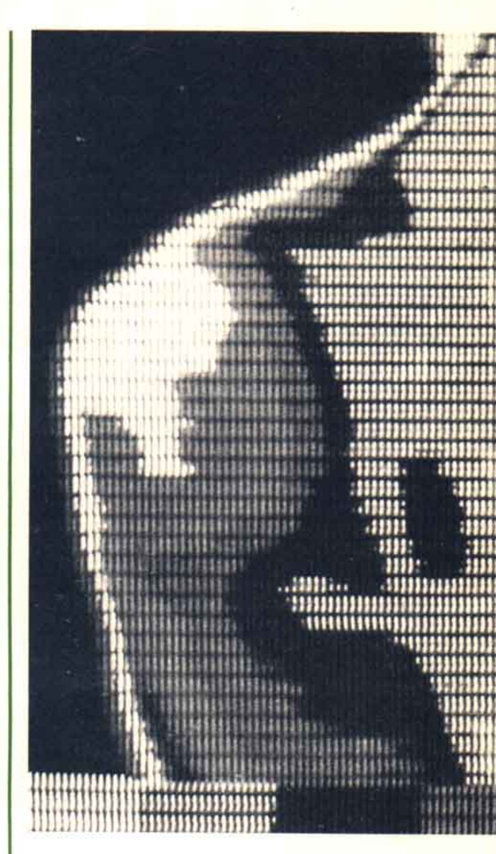
primer octeto de la tabla + n.° de carácter × 8

Una vez entendidos los conceptos de «color de tinta», «color de fondo», y el funcionamiento de la tabla 7, la tabla 6 no presenta ninguna dificultad.

La tabla 6 es la más pequeña de todas, con sólo 32 octetos de longitud. En nuestros ordenadores MSX comienza en el octeto 8192 y termina en el octeto 8223. Su función es almacenar los códigos de color de los caracteres que pueden salir en la pantalla. Esta tabla resulta mucho más cómoda de manejar si se tratan los números que se hallan en los octetos en el sistema numérico hexadecimal, o de base 16. Esto es lógico, pues son 16 también los colores de los que pueden disponer los ordenadores de la norma MSX:

&H0 = Transparente
&H1 = Negro
&H2 = Verde
&H3 = Verde claro
&H4 = Azul oscuro
&H5 = Azul claro
&H6 = Rojo oscuro
&H7 = Azul celeste
&H8 = Rojo
&H9 = Rojo claro
&HA = Amarillo oscuro
&HB = Amarillo claro
&HC = Verde oscuro
&HD = Magenta
&HE = Gris
&HF = Blanco

(Esta lista de los colores MSX seguro

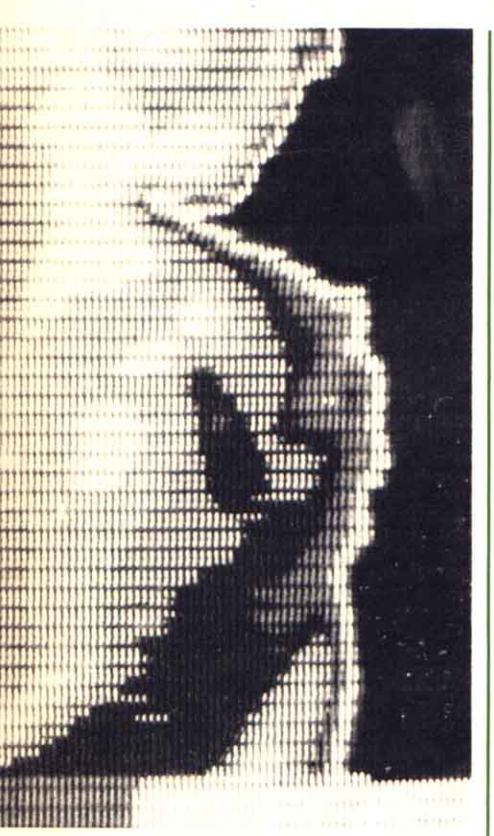


que se halla en las instrucciones de tu ordenador, pero conviene tenerla a mano con sus valores hexadecimales para las aplicaciones de la tabla 6).

Si tomamos en grupos de ocho *bit* los 256 caracteres de MSX, tendremos un total de 32 grupos, que se corresponden con los 32 octetos de la tabla 6. Así, los caracteres del 0 al 7 se rigen por el octeto 8192, los caracteres del 8 al 15 por el 8193,

los del 16 al 23 por el 8194, y continúa en esta sucesión hasta el octeto 8223, que rigen los caracteres del 248 al 255. La fórmula para hallar el octeto sería la siguiente: primer octeto de la tabla + INT (n.° de carácter/ 8).

Entendámos, no es que los octetos «rijan» a los caracteres. Cada octeto alamacena un número decimal entre 0 y 255, es decir, un número hexadecimal de dos cifras. La primera cifra hexadecimal representa el color de tinta, y la segunda el color



de fondo que han de llevar los caracteres pertenecientes al intervalo que rija ese octeto.

De esta manera, cuando en *SCREEN* 1 tecleamos en el ordenador COLOR 1,15
es lo mismo que si tecleáramos
FOR T=8192 TO 8223: VPOKE T, &H1F: NEXT

La única diferencia es que EN ESTE CASO la sentencia es más útil que *VPOKE*, porque lo hace más rápido. Sin embargo, en cualquier otro caso es mejor utilizar la memoria de vídeo, pues gracias a ella puedes lograr distintos colores del fondo y de la tinta. Si ahora dejas en tu ordenador las letras negras sobre fondo blanco y quieres que siga en esa combinación de colores en todas las letras excepto en las mayúsculas, puedes hacer que las mayúsculas salgan de otro color, por ejemplo con letras blancas sobre fondo azul oscuro (combinación &HF4). Apliquemos la fórmula anterior para la primera letra mayúscula, la «A» (carácter 65):

8192 + INT (52/8) =8200
y apliquémosla de nuevo para la «Z» (carácter 90):
8192 + INT (90/8) = 8203

Según estos resultados, las letras mayúsculas están regidas por los octetos 8200, 8201, 8202 y 8203. Cambiémosles el color:

FOR T=8200 TO 8203: *VPOKE* T, &HF4: NEXT

A partir de este momento, las letras mayúsculas, corchetes, signo de potencia, de división de enteros, de subrayado y el carácter « » aparecerán ahora en la pantalla con tinta blanca sobre fondo azul. Todos ellos han sido afectados por el cambio de octetos.

Las tablas 6 y 7 se encargan de realizar todas las comprobaciones del color de la pantalla.

Las tablas 5, 6 y 7 en la práctica

Dentro del gran número de aplicaciones que tienen estos dos tablas, cabe destacar tres de ellas.

La primera es la más sencilla, y consiste en escribir en pantalla textos de diversos colores. Pero al utilizarla te vas a encontrar muy pronto con un grave problema: habrá una misma letra que debe aparecer en los dos textos de un color diferente cada uno. Esto es muy fácil de resolver: el patrón de esa letra se puede copiar, gracias a la tabla 7, en otro carácter diferente (que dependa de otro octeto) y así habrá dos caracteres iguales, uno para un texto y otro para el otro texto. En realidad ya hay dos caracteres establecidos que son exactamente iguales, a pesar de hallarse los dos dentro de la tabla de caracteres MSX: el carácter nulo (n.° 0) y el espacio (n.° 32). Ambos poseén un patrón exactamente igual, que dispone que todos sus *píxels* sean impregnados del color de fondo, sin que ninguno de ellos pueda pintarse del color de la tinta. Precisamente por esa razón fue por la que los diseñadores del estándar MSX dispusieron que cuando el programador introdujera el mandato «CLS», la pantalla se llenará de caracteres de espacio, dando la impresión de que se ha «borrado» la pantalla, cuando en realidad no ha sufrido más que una sustitución de los caracteres que tenía por caracteres de espacio. (CLS = CLear Screen = Borrar Pantalla).

La segunda aplicación importante es la formación de gráficos en la pantalla en modo *SCREEN* 1, que puede ser posible formando los caracteres adecuado en la tabla 7, situándolos en la pantalla con la tabla 5 e impregnándolos de color con la tabla 6. Gracias al uso combinado de estas tres tablas se puden lograr desde rectángulos ó triángulos hasta complicadas formas curvadas, pasando por dibujos sencillos, creaciones plásticas y retratos. Recuerda que tienes un total de 32 combinaciones diferentes de colores.

Puedes también adornar los dibujos con las figuras móviles, de cuyo almacenamiento en la memoria de vídeo se hablará en próximos números.

Casi sería viable decir que el modo *SCREEN* 1 puede llegar a funcionar como el modo *SCREEN* 2, pero hay una diferencia decimal fundamental, y es que éste modo admite una combinación de color de tinta y color de fondo por cada cuadro de 1 fila × 8 columnas de *píxels*, con lo cual las posiblidades gráficas se multiplican, aparte de que el modo *SCREEN* 1 sólo puede efectuar dibujos de un máximo de 256 caracteres diferentes, mientras que el modo *SCREEN* 2 puede dibujar 768 cuadros con patrones diferentes y diversos colores. De todas maneras ya concretaremos más sobre el modo *SCREEN* 2 en el próximo número.

La tercera gran aplicación de las tablas 5, 6 y 7 es el movimiento de colores. Este programa puede ser un buen ejemplo de ello:

```
10 SCREEN 1: KEY OFF: COLOR, 15:
FOR R=1 TO 300:NEXT
20 VPOKE 8196, &H2F
30 FOR T=256 TO 263:VPOKE
T,255: FOR R=1 TO 50:NEXT:NEXT
40 COLOR 2,2:FOR T=256 TO 263:
VPOKE T,0:NEXT:COLOR 15
```

El objetivo de este programa es lograr intercambiar los colores de fondo y de tinta, de tal manera que el que antes era color de tinta lo sea ahora de fondo, y viceversa. Para ello hemos tomado los colores blanco y verde.

Esto sería muy fácil de hacer, sólo consistiría en un programa de dos órdenes: «COLOR 2,15», para establecer los colores iniciales, y «COLOR 15,2» para cambiarlos, pero lo que pretendemos es que éste cambio de color se realice por medio de un efecto estético que suavice la brusquedad que traería consigo un rápido cambio de color. El efecto consiste en que la pantalla se divida en rectángulos que se llenen lentamente del nuevo color de fondo.

En la línea 10 se borra la pantalla sin necesidad del mandato *CLS*, sólo con el mandato *SCREEN* 1 (como antes indicamos, no se borra la pantalla, sino que se llena de caracteres de espacio) y a continuación se establece el color blanco de fondo. La línea 10 incluye también un bucle *FOR-NEXT* retardador para que el observador pueda distinguir bien el color de fondo inicial. En la línea 20 se usa la tabla 6 para el carácter 32, que está regido por el octeto 8196 (8192 + INT (32/8 = 8196). A este octeto se le introduce la información que disponga que los espacios y los otros caracteres que rijan sean impregnados de color de fondo blanco y color de tinta verde. En la línea 30 y utilizando la tabla 7 se varía el patrón del carácter del espacio. Este patrón está situado a partir del octeto 256 (0 + 32 × 8 = 256) y termina 7 octetos más allá, en el 263. Los cambios que se efectúan en este patrón constituyen lo principal del programa, pues de ser un carácter lleno de ceros (*píxels* del color de fondo) pasa a ser un carácter lleno de unos (*píxels* del color de la tinta). Esta sustitución de ceros por unos se efectúa gradualmente, línea por línea, comenzando por la línea superior y terminando en la inferior, gracias al bucle *FOR-NEXT* de variable T. El bucle *FOR NEXT* de variable R es un bucle retardador, y es necesaria su inclusión en el programa, pues de lo contrario la situación se consumaría con demasiada velocidad para que el ojo humano pudiera captarlo.

Es posible realizar el efecto flash dentro del modo de pantalla SCREEN 1.

La sustitución de ceros por unos es el factor que determina la apariencia de los rectángulos. Si os preocupais en contrar los rectángulos, descubrireís que hay un total de 24, uno por cada línea de texto. Al irse llenando de color de tinta el patrón del espacio, se van llenando de tinta los espacios de la pantalla, con lo cual se llena de tinta toda la pantalla.

Lógicamente, ahora tendremos que dejar el carácter del espacio tal y como estaba al principio, lleno de ceros. Pero si llenamos de ceros el patrón inmediatamente después de la situación, volverá a verse blanca la pantalla, a causa de que el color de fondo sigue siendo blanco. Para evitar que al observador le parezca que tras llenar la pantalla de verde se vuelve bruscamente blanca, establecemos que el color de fondo en vez de ser blanco sea verde, como el color de tinta. En esta situación —color de fondo verde y color de tinta verde— cualquier escritura, dibujo o movimiento que haya en la pantalla será absolutamente invisible para el ojo humano. Esta es la situación que aprovechamos para llenar de ceros el patrón del espacio, dejándolos como siempre. Ahora, una vez que se ha terminado el proceso de situar al verde como color de fondo, sólo queda situar al blanco como color de tinta («COLOR 15»), y el programa llega a su fin.

Otro ejemplo del movimiento de colores es el efecto *FLASH*, del que ya hablamos en el número anterior. En el modo *SCREEN* 0 el efecto *FLASH* está limitado por los dos únicos colores que pueden aparecen en pantalla como color de fondo y color de tinta, por eso requiere la creación de caracteres negativos. Al ser, sin embargo, mucho mejor el tratamiento del color en el modo *SCREEN* 1, no hace falta nada de esto. Sólo bastaría con este programa:

```
10 SCREEN 1:KEY OFF:COLOR
15,8,8:LOCATE 15,7:PRINT «MSX»
20 FOR T=8201 TO 8203:VPOKE T,
&HF4:NEXT:FOR R=1 TO 200:NEXT
30 FOR T  8201 TO 8203:VPOKE T,
&H4F:NEXT:FOR R=1 TO 200:NEXT:
GOTO 20
```

Este programa comienza con un borrado de pantalla, estableciendo

tinta blanca sobre fondo verde y bordes rojos. Tras esto, la línea 10 imprime en la pantalla las letras «MSX» (FORZOSAMENTE MAYUSCULAS) a partir del cuadro de la línea 7 y columna 15. En la línea 20 se determina tinta blanca sobre fondo azul para los caracteres regidos por los octetos 8201, 8202 y 8203, entre los que se incluyen «M», «S» y «X», y en la 30 se determina tinta azul y fondo blanco sobre esos mismos caracteres. En ambas líneas se incluye también un bucle retardador de variable R. En último lugar, la línea 30 posee un *GOTO* 20, con lo que el efecto *FLASH* se producirá constantemente hasta que decidamos parar el programa con las teclas de *CONTROL* y *STOP*.

Esta combinación blanco-azul oscuro puede ser sustituida por cualquier otra, combinando adecuadamente las líneas 20 y 30.

Los colores de los caracteres que aparecen en TV, están guardados en la tabla 6.

En *SCREEN* 1 es posible situar varios efectos *FLASH* a la vez en la pantalla y de distintos colores, lo que proporcionará vistosidad a la presentación de cualquier programa.

En *SCREEN* 1 existen otras dos tablas, la 8 y la 9, pero éstas se refieren a las figuras móviles, también conocidas como «Sprites», y ambas tablas serán tratadas en el capítulo dedicado a las figuras móviles en todos los modos *SCREEN*.

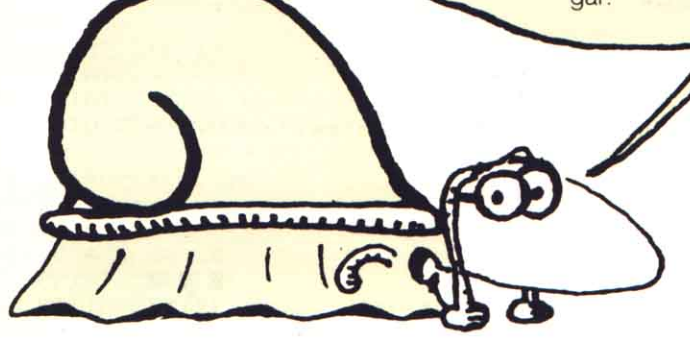
En el próximo número trataremos el sistema de almacenamiento de datos del modo *SCEEN* 2, también conocido como el modo de gráficos de gran definición, que contiene las tablas 10, 11, 12, 13, y 14. Gracias a ellas será posible trazar dibujos casi instantáneos, y también alcanzar el efecto *FLASH*.

José Mª Cavanillas

Aclaración

El mes pasado, en el artículo «la matemática y el ordenador», omitimos el programa que listamos a continuación.

Este, completa lo expuesto en dicho artículo y servirá como muestra a la hora de optimizar el ordenador y utilizarlo para algo más que para jugar.



```
10 'ECUACIONES ALGEBRAICAS - JAFC
20 'VERSION  15.08.85 - 3414 bytes
30 '==============================
40 '
50 SCREEN 0,,0:WIDTH 39:KEY OFF:DEFINT I-N
60 KEY2,"GOTO 1570"+CHR$(13)
70 KEY3,"GOTO 2070"+CHR$(13)
80 CLS:GOSUB 4040
90 COLOR 1,2,2
100 PRINT "  ┌───────────────────────────────────┐"
110 PRINT "  │";SPC(35);"│"
120 PRINT "  │       ECUACIONES ALGEBRAICAS.     │"
130 PRINT "  │";SPC(35);"│"
140 PRINT "  ┌───────────────────────────────────┐"
150 FOR I=1 TO 11
160 PRINT "  │";SPC(35);"│"
170 NEXT I
180 PRINT "  └───────────────────────────────────┘"
190 A$(0)="INFORMACION"
200 A$(1)="ECUACION DE 2º GRADO"
210 A$(2)="REGULA FALSI"
220 A$(3)="METODO DE NEWTON"
230 FOR I0=0 TO 3
240 LOCATE 6,7+2*I0
250 PRINT I0;": ";A$(I0)
260 NEXT I0
270 LOCATE 2,20
280 PRINT "ELEGIR OPCION"
290 LOCATE 30,22
300 PRINT "█i█"
310 W$=INPUT$(1)
320 IF W$="0" OR W$="1" OR W$="2" OR W$="3" THEN 350
330 IF ASC(W$)=27 THEN CLS:COLOR 15,4,4:END
340 GOTO 310
350 I0=VAL(W$)
360 CLS
370 COLOR 15,4,4
380 PRINT TAB(2)"┌";
390 FOR I=1 TO LEN(A$(I0))
400 PRINT "─";
410 NEXT I
420 PRINT "┐"
```

```
430 PRINT TAB(2)"|";A$(I0);"|"
440 PRINT TAB(2)"└";
450 FOR I=1 TO LEN(A$(I0))
460 PRINT "-";
470 NEXT I
480 PRINT "┘"
490 PRINT
500 ON I0+1 GOTO 4540,1000,1540,2010
1000 '
1010 'EC. DE 2º GRADO
1020 '===============
1030 '
1040 PRINT "  Ax²+Bx+C=0",," Introducir los coeficientes:":PRINT
1050 INPUT "  A,B,C";A,B,C:PRINT
1060 B=-B/2/A:C=C/A
1070 D=B^2-C
1080 IF D<0 THEN 1140
1090 D=SQR(D)
1100 X1=B+D:X2=B-D
1110 PRINT "  RAICES REALES:";USING "X1=#####.####";X1
1120 PRINT TAB(16);USING "X2=#####.####";X2
1130 GOTO 2540
1140 D=SQR(-D)
1150 PRINT "  RAICES IMAGINARIAS:"
1160 PRINT USING "  X1=#####.####   +i•#####.####";B;D
1170 PRINT USING "  X2=#####.####   -i•#####.####";B;D
1180 GOTO 2540
1500 '
1510 'REGULA FALSI
1520 '============
1530 '
1540 PRINT " Definir función en línea 3540"," Pulsar F2"
1550 PRINT
1560 LIST 3540
1570 GOSUB 4040:I0=2
1580 PRINT
1590 INPUT "INTERVALO:(a,b)";A,B
1600 PRINT
1610 INPUT "Nº de cifras significativas";N
1620 I=0:U=1E+60
1630 X=A:GOSUB 3540:C=Y
1640 X=B:GOSUB 3540:D=Y
1650 I=I+1:X=(A*D-B*C)/(D-C):GOSUB 3540
1660 IF ABS((X-U)/X)<10^-N GOTO 1700
1670 U=X
1680 IF SGN(Y)=SGN(D) THEN B=X:D=Y:GOTO     1650
1690 A=X:C=Y:GOTO 1650
1700 PRINT
1710 PRINT "X=";X
1720 PRINT
1730 PRINT "Y=";Y
1740 PRINT
1750 PRINT "Nº de iteraciones=";I
1760 GOTO 2540
2000 '
2010 'METODO DE NEWTON
2020 '================
2030 '
2040 PRINT " Definir función en línea 3540"," Definir derivada en línea 3560","
Pulsar F3"
2050 PRINT
2060 LIST 3540-3560
2070 GOSUB 4040:I0=3
2080 PRINT
```

```
2090 INPUT "ABSCISA INICIAL:a";X
2100 PRINT
2110 INPUT "Nº de cifras significativas";N
2120 GOSUB 3540:X1=X-Y/Y1
2130 IF ABS((X1-X)/X1)>=10^-N THEN X=X1:I=I+1:GOTO 2120
2140 PRINT
2150 PRINT "X=";X1
2160 X=X1:GOSUB 3540
2170 PRINT
2180 PRINT "Y=";Y
2190 PRINT
2200 PRINT "Nº de iteraciones";I
2210 GOTO 2540
2500 '
2510 'Bifurcaciones
2520 '-------------
2530 '
2540 LOCATE 30,22
2550 PRINT "■ë■ï■"
2560 W$=INPUT$(1)
2570 IF ASC(W$)=24 THEN 80
2580 IF ASC(W$)=27 THEN CLS:END
2590 GOTO 2560
3000 '
3010 'Subrutina de error
3020 '-----------------
3030 '
3040 IF ERR=11 AND I0=2 OR I0=3 THEN PRINT:PRINT "División por 0. No puedo conti
nuar.":IF I0=2 THEN RESUME 1700 ELSE RESUME 2140
3050 CLS:LOCATE15,11:PRINT "ERROR";ERR:RESUME 2540
3500 '
3510 'A definir por usuario
3520 '--------------------
3530 '
3540 Y=SIN(X)
3550 IF I0=2 THEN RETURN
3560 Y1=COS(X)
3570 RETURN
4000 '
4010 'Inicialización
4020 '--------------
4030 '
4040 ON ERROR GOTO 3040
4050 DEF FNA(A$)=2048+ASC(A$)*8
4060 A$="E":AW(1)=FNA(A$)
4070 A$="ï":BW(1)=FNA(A$)
4080 A$="S":AW(2)=FNA(A$)
4090 A$="ë":BW(2)=FNA(A$)
4100 FOR I=1 TO 2
4110 FOR J=0 TO 7
4120 VPOKE BW(I)+J,255-VPEEK(AW(I)+J)
4130 NEXTJ
4140 NEXT I
4150 RETURN
4500 '
4510 'INFORMACION
4520 '===========
4530 '
4540 PRINT "  KEY 1= motor + [RETURN]"
4550 PRINT "  KEY 2= goto 1060 + [RETURN]"
4560 PRINT "  KEY 3= goto 1560 + [RETURN]"
4570 PRINT "  ■ï■  = [ESC]:Finaliza el programa."
4580 PRINT "  ■ë■  = [SELECT]:Vuelve a MENU."
4590 GOTO 2540
```

Ensamblador/desensamblador Philips



Una de las utilidades casi imprescindibles para programar en lenguaje máquina es un ensamblador. Esto se debe a que sin su ayuda la tarea de programación se convierte en una labor tediosa y propensa a generar errores, ya que todo se introduce como códigos hexadecimales.

En cambio el uso de un ensamblador nos permite introducir los nemotécnicos (nombre de las intrucciones) en lugar de los códigos anteriores y hacer uso de etiquetas para referenciar las direcciones, con lo que el ordenador se encarga de este cálculo.

En este apartado iremos examinando mes a mes los distintos en-

Tabla de instrucciones del Z-80

En la tabla que viene a continuación se dan todas las instrucciones que posee el Z-80, separadas según los distintos tipos de direccionamiento. Cada uno tiene tres columnas que indican lo siguiente: La primera es el nombre de la instrucción tal como se debe poner en el ensambador (si utilizamos alguna utilidad de este tipo). En este caso las letras minúsculas deben sustituirse por números de uno o dos bytes. Todos los «dis», son desplazamientos de un byte, los «ddd» indican un dato de un byte, los «dddd» son datos byte y «ADDR» (en mayúsculas) es una dirección de dos bytes.

En la segunda columna se da el formato de dicha instrucción, pero en hexadecimal. En este caso cada «xx» es un byte numérico, por lo que habrá que sustituirlo por su valor a la hora de teclear. Hay que tener en cuenta que las direcciones de dieciséis bits deben descomponerse en dos y poner la parte inferior primero.

La última columna indica el número de ciclos de reloj que tarda en ejecutarse cada instrucción y sirve para hacer cálculos del tiempo que tarda en ejecutarse un programa o una rutina, para lo que basta sumar lo que tarda cada instrucción y saber que la máquina tiene un ciclo de reloj de 3,5 megahercios, es decir en cada segundo hay tres millones y medio de ciclos de reloj.

La tabla se puede usar, por tanto, como recordatorio de las instrucciones existentes y como indicador de los códigos hexadecimales si es que se ensambla a mano, por lo que conviene tenerla siempre cerca.



Nombre	Hexadecimal	Ciclos	Nombre	Hexadecimal	Ciclos
ADC A,(HL)	8E	7	AND A	A7	4
ADC A,(IX+dis)	DD 8E XX	19	AND B	A0	4
ADC A,(IY+dis)	FD 8E XX	19	AND C	A1	4
ADC A,A	8F	4	AND D	A2	4
ADC A,B	88	4	AND ddd	E6 dd	7
ADC A,C	89	4	AND E	A3	4
ADC A,D	8A	4	AND H	A4	4
ADC A,ddd	CE dd	7	AND L	A5	4
ADC A,E	8B	4	BIT 0,(HL)	CB 46	12
ADC A,H	8C	4	BIT 0,(IX+dis)	DD CB XX 46	20
ADC A,L	8D	4	BIT 0,(IY+dis)	FD CB XX 46	20
ADC HL,BC	ED 4A	15	BIT 0,A	CB 47	8
ADC HL,DE	ED 5A	15	BIT 0,B	CB 40	8
ADC HL,HL	ED 6A	15	BIT 0,C	CB 41	8
ADC HL,SP	ED 7A	15	BIT 0,D	CB 42	8
ADD A,(HL)	86	7	BIT 0,E	CB 43	8
ADD A,(IX+dis)	DD 86 XX	19	BIT 0,H	CB 44	8
ADD A,(IY+dis)	FD 86 XX	19	BIT 0,L	CB 45	8
ADD A,A	87	4	BIT 1,(HL)	CB 4E	12
ADD A,B	80	4	BIT 1,(IX+dis)	DD CB XX 4E	20
ADD A,C	81	4	BIT 1,(IY+dis)	FD CB XX 4E	20
ADD A,D	82	4	BIT 1,A	CB 4F	8
ADD A,ddd	C6 dd	7	BIT 1,B	CB 48	8
ADD A,E	83	4	BIT 1,C	CB 49	8
ADD A,H	84	4	BIT 1,D	CB 4A	8
ADD A,L	85	4	BIT 1,E	CB 4B	8
ADD HL,BC	09	11	BIT 1,H	CB 4C	8
ADD HL,DE	19	11	BIT 1,L	CB 4D	8
ADD HL,HL	29	11	BIT 2,(HL)	CB 56	12
ADD HL,SP	39	11	BIT 2,(IX+dis)	DD CB XX 56	20
ADD IX,BC	DD 09	15	BIT 2,(IY+dis)	FD CB XX 56	20
ADD IX,DE	DD 19	15	BIT 2,A	CB 57	8
ADD IX,IX	DD 29	15	BIT 2,B	CB 50	8
ADD IX,SP	DD 39	15	BIT 2,C	CB 51	8
ADD IY,BC	FD 09	15	BIT 2,D	CB 52	8
ADD IY,DE	FD 19	15	BIT 2,E	CB 53	8
ADD IY,IY	FD 29	15	BIT 2,H	CB 54	8
ADD IY,SP	FD 39	15	BIT 2,L	CB 55	8
AND (HL)	A6	7	BIT 3,(HL)	CB 5E	12
AND (IX+dis)	DD A6 XX	19	BIT 3,(IX+dis)	DD CB XX 5E	20
AND (IY+dis)	FD A6 XX	19	BIT 3,(IY+dis)	FD CB XX 5E	20

sambladores existentes, de modo que se pueda elegir entre todos ellos.

El primero que probamos es distribuidor por la casa *PHILIPS*, aunque vale para cualquier ordenador MSX. Viene presentado en cinta y acompañado de un manual con una gran extensión de páginas.

El proceso de cargo es algo complicado ya que previamente a la carga (con *BLOAD*, hay que hacer un *CLEAR*. Este proceso podría haberse arreglado con un pequeño programa, *BASIC* que lo hiciera automáticamente y ahorrara teclearlo, aunque el fabriante ha preferido dejarlo en nuestras manos.

Una ez cargado nos aparece el mensaje *ZEN* que indica que está dispuesto ha aceptar nuestras ordenes. Estas nos permiten introducir el texto fuente por medio de un editor, salvarlo en cinta y volver a leerlo y ensamblarlo. También están las instrucciones destinadas al desensamblado y testeado de programas.

En el editor existe lo que podríamos llamar numeración dinámica, ya que el programa se encarga de nu-

Nombre	Hexadecimal	Ciclos
BIT 3,A	CB 5F	8
BIT 3,B	CB 58	8
BIT 3,C	CB 59	8
BIT 3,D	CB 5A	8
BIT 3,E	CB 5B	8
BIT 3,H	CB 5C	8
BIT 3,L	CB 5D	8
BIT 4,(HL)	CB 66	12
BIT 4,(IX+dis)	DD CB XX 66	20
BIT 4,(IY+dis)	FD CB XX 66	20
BIT 4,A	CB 67	8
BIT 4,B	CB 60	8
BIT 4,C	CB 61	8
BIT 4,D	CB 62	8
BIT 4,E	CB 63	8
BIT 4,H	CB 64	8
BIT 4,L	CB 65	8
BIT 5,(HL)	CB 6E	12
BIT 5,(IX+dis)	DD CB XX 6E	20
BIT 5,(IY+dis)	FD CB XX 6E	20
BIT 5,A	CB 6F	8
BIT 5,B	CB 68	8
BIT 5,C	CB 69	8
BIT 5,D	CB 6A	8
BIT 5,E	CB 6B	8
BIT 5,H	CB 6C	8
BIT 5,L	CB 6D	8
BIT 6,(HL)	CB 76	12
BIT 6,(IX+dis)	DD CB XX 76	20
BIT 6,(IY+dis)	FD CB XX 76	20
BIT 6,A	CB 77	8
BIT 6,B	CB 70	8
BIT 6,C	CB 71	8
BIT 6,D	CB 72	8
BIT 6,E	CB 73	8
BIT 6,H	CB 74	8
BIT 6,L	CB 75	8
BIT 7,(HL)	CB 7E	12
BIT 7,(IX+dis)	DD CB XX 7E	20
BIT 7,(IY+dis)	FD CB XX 7E	20
BIT 7,A	CB 7F	8

Nombre	Hexadecimal	Ciclos
BIT 7,B	CB 78	8
BIT 7,C	CB 79	8
BIT 7,D	CB 7A	8
BIT 7,E	CB 7B	8
BIT 7,H	CB 7C	8
BIT 7,L	CB 7D	8
CALL ADDR	CD XX XX	17
CALL C,ADDR	DC XX XX	17/10
CALL M,ADDR	FC XX XX	17/10
CALL NC,ADDR	D4 XX XX	17/10
CALL NZ,ADDR	C4 XX XX	17/10
CALL P,ADDR	F4 XX XX	17/10
CALL PE,ADDR	EC XX XX	17/10
CALL PO,ADDR	E4 XX XX	17/10
CALL Z,ADDR	CC XX XX	17/10
CCF	3F	4
CP (HL)	BE	7
CP (IX+dis)	DD BE XX	19
CP (IY+dis)	FD BE XX	19
CP A	BF	4
CP B	B8	4
CP C	B9	4
CP D	BA	4
CP ddd	FE dd	7
CP E	BB	4
CP H	BC	4
CP L	BD	4
CPD	ED A9	16
CPDR	ED B9	21/16
CPI	ED A1	16
CPIR	ED B1	21/16
CPL	2F	4
DAA	27	4
DEC (HL)	35	11
DEC (IX+dis)	DD 35 XX	23
DEC (IY+dis)	FD 35 XX	23
DEC A	3D	4
DEC B	05	4
DEC BC	0B	6
DEC C	0D	4
DEC D	15	4

merar las líneas automáticamente y modificar esta numeración en caso necesario. Otro aspecto importante es que existe una línea actual, en la que se realizan todas las instrucciones que se tecleen. Así por ejemplo, cuando se pulsa Z1 *RETURN*, se borra la línea actual. Si en vez de Z1, se hubiese puesto Z2, se borraría la línea actual y la siguiente, quedando como nueva linea actual la que hubiese detrás de estas dos y cambiando la numeración en función de esto. Esto puede resultar algo peligroso, ya que las líneas no se ven constantemente en pantalla, si no que hay que darle la orden «P» (seguida opcionalmente por un número que indica las líneas a mostrar) para que salga un listado y poder ver cual es a línea actual y las posteriores a ella.

Entre los diversos comandos que se pueden usar, están los siguientes: «E» nos permite introducir una nueva línea antes de la actual, desplazándose la numeración de ésta si fuese necesario.

«U» sube una línea, haciendo que la línea actual sea la anterior a la que

Nombre	Hexadecimal	Ciclos
DEC DE	1B	6
DEC E	1D	4
DEC H	25	4
DEC HL	2B	6
DEC IX	DD 2B	10
DEC IY	FD 2B	10
DEC L	2D	4
DEC SP	3B	6
DI	F3	4
DJNZ,dis	10 XX	13/8
EI	FB	4
EX (SP),HL	E3	19
EX (SP),IX	DD E3	23
EX (SP),IY	FD E3	23
EX AF,AF'	08	4
EX DE,HL	EB	4
EXX	D9	4
HLT	76	4
IM 0	ED 46	8
IM 1	ED 56	8
IM 2	ED 5E	8
IN A,(C)	ED 78	12
IN A,port	DB XX	11
IN B,(C)	ED 40	12
IN C,(C)	ED 48	12
IN D,(C)	ED 50	12
IN E,(C)	ED 58	12
IN H,(C)	ED 60	12
IN L,(C)	ED 68	12
INC (HL)	34	11
INC (IX+dis)	DD 34 XX	23
INC (IY+dis)	FD 34 XX	23
INC A	3C	4
INC B	04	4
INC BC	03	6
INC C	0C	4
INC D	14	4
INC DE	13	6
INC E	1C	4
INC H	24	4
INC HL	23	6

Nombre	Hexadecimal	Ciclos
INC IX	DD 23	10
INC IY	FD 23	10
INC L	2C	4
INC SP	33	6
IND	ED AA	16
INDR	ED BA	21/16
INI	ED A2	16
INIR	ED B2	21/16
JP (HL)	E9	4
JP (IX)	DD E9	8
JP (IY)	FD E9	8
JP ADDR	C3 XX XX	10
JP C,ADDR	DA XX XX	10
JP M,ADDR	FA XX XX	10
JP NC,ADDR	D2 XX XX	10
JP NZ,ADDR	C2 XX XX	10
JP P,ADDR	F2 XX XX	10
JP PE,ADDR	EA XX XX	10
JP PO,ADDR	E2 XX XX	10
JP Z,ADDR	CA XX XX	10
JR C,dis	38 XX	12
JR dis	18 XX	12/7
JR NC,dis	30 XX	12/7
JR NZ,dis	20 XX	12/7
JR Z,dis	28 XX	12/7
LD (ADDR),A	32 XX XX	13
LD (ADDR),BC	ED 43 XX XX	20
LD (ADDR),DE	ED 53 XX XX	20
LD (ADDR),HL	ED 63 XX XX	20
LD (ADDR),HL	22 XX XX	16
LD (ADDR),IX	DD 22 XX XX	20
LD (ADDR),IY	FD 22 XX XX	20
LD (ADDR),SP	ED 73 XX XX	20
LD (BC),A	02	7
LD (DE),A	12	7
LD (HL),A	77	7
LD (HL),B	70	7
LD (HL),C	71	7
LD (HL),D	72	7
LD (HL),ddd	36 dd	10
LD (HL),E	73	7

estábamos. Del mismo modo «D» baja una línea. En ambos casos se puede poner detrás un número que indique las líneas a bajar y subir.

«H» nos muestra el tamaño del fichero de texto y la memoria que queda libre.

La «K» borra todo el fichero, pudiéndose considerar como el equivalente del *NEW* del *BASIC*.

Un comando muy bersatil es la «L», que busca una serie de caracteres en todo el texto y se para en la línea que los contenga. Esto posibilita los desplazamientos rápidos a sitios específicos como son las etiquetas, instrucciones determinadas, etc.

La «N» es abreviatura de *NEW*, pero al contrario de esta instrucción del *BASIC*, en este caso lo que se nos permite es modificar la línea actual dejando lo que esta bien y cambiando los datos erroneos.

La «T» convierte en línea actual la que se indica después de esta letra. A diferencia de «U» y «D», que actúan relativamente a la línea en curso, este comando se mueve con respecto a la primera línea existente, por lo que conviene hacer un listado antes

Nombre	Hexadecimal	Ciclos	Nombre	Hexadecimal	Ciclos
LD (HL),H	74	7	LD B,ddd	06 dd	7
LD (HL),L	75	7	LD B,E	43	4
LD (IX+dis),A	DD 77 XX	19	LD B,H	44	4
LD (IX+dis),B	DD 70 XX	19	LD B,L	45	4
LD (IX+dis),C	DD 71 XX	19	LD BC,(ADDR)	ED 4B XX XX	20
LD (IX+dis),D	DD 72 XX	19	LD BC,dddd	01 dd dd	10
LD (IX+dis),ddd	DD 36 XX dd	19	LD C,(HL)	4E	7
LD (IX+dis),E	DD 73 XX	19	LD C,(IX+dis)	DD 4E XX	19
LD (IX+dis),H	DD 74 XX	19	LD C,(IY+dis)	FD 4E XX	19
LD (IX+dis),L	DD 75 XX	19	LD C,A	4F	4
LD (IY+dis),A	FD 77 XX	19	LD C,B	48	4
LD (IY+dis),B	FD 70 XX	19	LD C,C	49	4
LD (IY+dis),C	FD 71 XX	19	LD C,D	4A	4
LD (IY+dis),D	FD 72 XX	19	LD C,ddd	0E dd	7
LD (IY+dis),ddd	FD 36 XX dd	19	LD C,E	4B	4
LD (IY+dis),E	FD 73 XX	19	LD C,H	4C	4
LD (IY+dis),H	FD 74 XX	19	LD C,L	4D	4
LD (IY+dis),L	FD 75 XX	19	LD D,(HL)	56	7
LD A,(ADDR)	3A XX XX	13	LD D,(IX+dis)	DD 56 XX	19
LD A,(BC)	0A	7	LD D,(IY+dis)	FD 56 XX	19
LD A,(DE)	1A	7	LD D,A	57	4
LD A,(HL)	7E	7	LD D,B	50	4
LD A,(IX+dis)	DD 7E XX	19	LD D,C	51	4
LD A,(IY+dis)	FD 7E XX	19	LD D,D	52	4
LD A,A	7F	4	LD D,ddd	16 dd	7
LD A,B	78	4	LD D,E	53	4
LD A,C	79	4	LD D,H	54	4
LD A,D	7A	4	LD D,L	55	4
LD A,ddd	3E dd	7	LD DE,(ADDR)	ED 5B XX XX	20
LD A,E	7B	4	LD DE,dddd	11 dd dd	10
LD A,H	7C	4	LD E,(HL)	5E	7
LD A,I	ED 57	9	LD E,(IX+dis)	DD 5E XX	19
LD A,L	7D	4	LD E,(IY+dis)	FD 5E XX	19
LD A,R	ED 5F	9	LD E,A	5F	4
LD B,(HL)	46	7	LD E,B	58	4
LD B,(IX+dis)	DD 46 XX	19	LD E,C	59	4
LD B,(IY+dis)	FD 46 XX	19	LD E,D	5A	4
LD B,A	47	4	LD E,ddd	1E dd	7
LD B,B	40	4	LD E,E	5B	4
LD B,C	41	4	LD E,H	5C	4
LD B,D	42	4	LD E,L	5D	4

Como carencia siginificativa podemos indicar la inexistencia de macros, que nos permiteen (cuando existen) definir una serie de instrucciones por un nombre y cuando hacemos refgerencia a este nombre, el programa lo sustituye por dichas instrucciones.

El programa funciona bien y no tienes pegas una vez que se aprende a manejarlo correctamente, lo que resulta algo difícil si no se posee experiencia previa en ensambladores, ya que aunque de un grosor elevado, el manual resulta excesivamente escuerto y sólo destina 11 páginas del total a explicar todas las instrucciones. Estando destinado el resto a incluir un listado de todas las instrucciones del Z80 y del propio ensamblador.

Como conclusión definitiva podríamos indicar que funciona ien y ofrece características interesantes, sobre todo teniendo en cuenta que está basado en cinta, echándose en falta una mayor claridad de manejo y un manual más amplio y con ejemplos explicativos.

Nombre	Hexadecimal	Ciclos	Nombre	Hexadecimal	Ciclos
LD D,(HL)	66	7	NOP	00	4
LD H,(IX+dis)	DD 66 XX	19	OR (HL)	B6	7
LD H,(IY+dis)	FD 66 XX	19	OR (IX+dis)	DD B6 XX	19
LD H,A	67	4	OR (IY+dis)	FD B6 XX	19
LD H,B	60	4	OR A	B7	4
LD H,C	61	4	OR B	B0	4
LD H,D	62	4	OR C	B1	4
LD H,ddd	26 dd	7	OR D	B2	4
LD H,E	63	4	OR ddd	F6 dd	7
LD H,H	64	4	OR E	B3	4
LD H,L	65	4	OR H	B4	4
LD HL,(ADDR)	ED 6B XX XX	20	OR L	B5	4
LD HL,(ADDR)	2A XX XX	16	OTDR	ED BB	21/16
LD HL,dddd	21 dd dd	10	OTIR	ED B3	21/16
LD I,A	ED 47	9	OUT (C),A	ED 79	12
LD IX,(ADDR)	DD 2A XX XX	20	OUT (C),B	ED 41	12
LD IX,dddd	DD 21 dd dd	14	OUT (C),C	ED 49	12
LD IY,(ADDR)	FD 2A XX XX	20	OUT (C),D	ED 51	12
LD IY,dddd	FD 21 dd dd	14	OUT (C),E	ED 59	12
LD L,(HL)	6E	7	OUT (C),H	ED 61	12
LD L,(IX+dis)	DD 6E XX	19	OUT (C),L	ED 69	12
LD L,(IY+dis)	FD 6E XX	19	OUT port,A	D3 port	11
LD L,A	6F	4	OUTD	ED AB	16
LD L,B	68	4	OUTI	ED A3	16
LD L,C	69	4	POP AF	F1	10
LD L,D	6A	4	POP BC	C1	10
LD L,ddd	2E dd	7	POP DE	D1	10
LD L,E	6B	4	POP HL	E1	10
LD L,H	6C	4	POP IX	DD E1	14
LD L,L	6D	4	POP IY	FD E1	14
LD R,A	ED 4F	9	PUSH AF	F5	11
LD SP,(ADDR)	ED 7B XX XX	20	PUSH BC	C5	11
LD SP,dddd	31 dd dd	10	PUSH DE	D5	11
LD SP,HL	F9	6	PUSH HL	E5	11
LD SP,IX	DD F9	10	PUSH IX	DD E5	15
LD SP,IY	FD F9	10	PUSH IY	FD E5	15
LDD	ED A8	16	RES 0,(HL)	CB 86	15
LDDR	ED B8	21/16	RES 0,(IX+dis)	DD CB XX 86	23
LDI	ED A0	16	RES 0,(IY+dis)	FD CB XX 86	23
LDIR	ED B0	21/16	RES 0,A	CB 87	8
NEG	ED 44	8	RES 0,B	CB 80	8

de saber la posición de cada línea.

Además de estas series comentadas existe otra que graba el fichero en disco, verifica que se ha grabado bien y lo lée.

Para ensamblar un texto que se encuentra en memoria, basta dar «A», a continuación de lo cual se nos pide dónde queremos generar un listado, si por pantalla, por impresora o a un dispositivo externo (un fichero en disco por ejemplo). Tras lo cual se realiza el proceso.

Además de las instrucciones de Z80 propiamente dichas, todo programa a ensamblar puede usar una serie de seudo operandos que le dan instrucciones al ensamblador de cómo funcionar y no generan código máquina. *LOAD* seguido de un número, indica la dirección inicial en donde se almacena el programa, *ORG*, también seguido de un número, indica que dirección se toma como base para calcular todos los saltos y referencia a direcciones. Salvo raras excepciones las direcciones indicadas en *LOAD* y *ORG* deben ser las mismas y deben indicarse al principio de todo. Otra instrucción que

Nombre	Hexadecimal	Ciclos	Nombre	Hexadecimal	Ciclos
RES 0,C	CB 81	8	RES 4,D	CB A2	8
RES 0,D	CB 82	8	RES 4,E	CB A3	8
RES 0,E	CB 83	8	RES 4,H	CB A4	8
RES 0,H	CB 84	8	RES 4,L	CB A5	8
RES 0,L	CB 85	8	RES 5,(HL)	CB AE	15
RES 1,(HL)	CB 8E	15	RES 5,(IX+dis)	DD CB XX AE	23
RES 1,(IX+dis)	DD CB XX 8E	23	RES 5,(IY+dis)	FD CB XX AE	23
RES 1,(IY+dis)	FD CB XX 8E	23	RES 5,A	CB AF	8
RES 1,A	CB 8F	8	RES 5,B	CB A8	8
RES 1,B	CB 88	8	RES 5,C	CB A9	8
RES 1,C	CB 89	8	RES 5,D	CB AA	8
RES 1,D	CB 8A	8	RES 5,E	CB AB	8
RES 1,E	CB 8B	8	RES 5,H	CB AC	8
RES 1,H	CB 8C	8	RES 5,L	CB AD	8
RES 1,L	CB 8D	8	RES 6,(HL)	CB B6	15
RES 2,(HL)	CB 96	15	RES 6,(IX+dis)	DD CB XX B6	23
RES 2,(IX+dis)	DD CB XX 96	23	RES 6,(IY+dis)	FD CB XX B6	23
RES 2,(IY+dis)	FD CB XX 96	23	RES 6,A	CB B7	8
RES 2,A	CB 97	8	RES 6,B	CB B0	8
RES 2,B	CB 90	8	RES 6,C	CB B1	8
RES 2,C	CB 91	8	RES 6,D	CB B2	8
RES 2,D	CB 92	8	RES 6,E	CB B3	8
RES 2,E	CB 93	8	RES 6,H	CB B4	8
RES 2,H	CB 94	8	RES 6,L	CB B5	8
RES 2,L	CB 95	8	RES 7,(HL)	CB BE	15
RES 3,(HL)	CB 9E	15	RES 7,(IX+dis)	DD CB XX BE	23
RES 3,(IX+dis)	DD CB XX 9E	23	RES 7,(IY+dis)	FD CB XX BE	23
RES 3,(IY+dis)	FD CB XX 9E	23	RES 7,A	CB BF	8
RES 3,A	CB 9F	8	RES 7,B	CB B8	8
RES 3,B	CB 98	8	RES 7,C	CB B9	8
RES 3,C	CB 99	8	RES 7,D	CB BA	8
RES 3,D	CB 9A	8	RES 7,E	CB BB	8
RES 3,E	CB 9B	8	RES 7,H	CB BC	8
RES 3,H	CB 9C	8	RES 7,L	CB BD	8
RES 3,L	CB 9D	8	RET	C9	10
RES 4,(HL)	CB A6	15	RET C	D8	11/5
RES 4,(IX+dis)	DD CB XX A6	23	RET M	F8	11/5
RES 4,(IY+dis)	FD CB XX A6	23	RET NC	D0	11/5
RES 4,A	CB A7	8	RET NZ	C0	11/5
RES 4,B	CB A0	8	RET P	F0	11/5
RES 4,C	CB A1	8	RET PE	E8	11/5

también es necesario poner es *END*, que indica el final del texto, su ausencia provocaría un error de ensamblado.

Otras instrucciones que incluye son *DW*, *DB* y *DS*, que reservan espacio para una palabra (dos *byte* u otro espacio distinto que se indica después de *DS*. Así mismo está el habitual *EQU* que nos permite asignar direcciones a las etiquetas, con lo que podemos hacer SUMA EQU 40500 y posteriormente hacer referencia a SUMA. Eso ya lo hemos hecho (a mano) en los listados de meses pasados y lo haremos en los siguientes, dando listados generados por este u otro ensamblador de modo que se disponga del listado hexadecimal del original.





Trucos

Muchos son los usarios que poséen el SV-328, buen ordenador aunque no es del todo compatible con el MSX de la familia, el SV-728. Aunque la sección está abierta a trucos, ideas etc., creemos oportuno publicar las modificaciones necesarias para poder adaptar la mayoría de los programas (siendo algunos difíciles de modificar).
Para empezar, destacremos que la principal diferencia entre estos ordenadores son los modos de pantalla, donde SCREEN 0 es el modo de textos, tanto en el MSX como en el SV-328. De forma que si un programa incorpora este comando, no será necesario modificarlo. Sin embargo, SCREEN 1 en el SV-328, corresponde al modo de alta resolución, mientras que en el MSX corresponde al modo de texto con sprites. Este modo sistuituye al modo 1 y 2 del MSX, ajustándonos a las siguientes normas:
1) Cuando en un programa MSX se encuentre la instrucción SCREEN 1 en el SV-328, cambiaremos todos los parámetros de las instrucciones LOCATE, multiplicados por 8. Por ejemplo:
Programa MSX:

```
10 SCREEN 1
20 LOCATE 10,10
30 CLS
40 PRINT "DEMOSTRACION"
50 GOTO 50
```

Programa SV-328:

```
10 SCREEN 1
20 LOCATE 10*8,10*8:'(80,80)
30 CLS
40 PRINT "DEMOSTRACION"
50 GOTO 50
```

2) Cuando en el programa MSX, se encuentre la instrucción SCREEN 2, en el SV-328 se sustituirá por SCREEN 1, teniendo en cuenta que:
a) las instrucciones OPEN «GRP»: AS # 1 deben eliminarse, y
b) cuando encuentre una instrucción PSET (X,Y) seguida por un PRINT # 1, la instrucción PSET se sustituirá por LOCATE, pero con los mismos parámetros, mientras que la instrucción PRINT# 1, se sustituirá por un PRINT.

Por ejemplo:

Programa MSX:

```
10 SCREEN 2 : color
20 OPEN "GRP:" AS #1
30 PSET (100,100)
40 PRINT#1, "PRUEBA"
50 CLOSE#1
60 GOTO 60
```

Programa SV-328:

```
10 SCREEN 1
20 ' ELIMINADA
30 LOCATE 100,100
40 PRINT "PRUEBA"
50 ' ELIMINADA
60 GOTO 60
```

Por último el equivalente del SCREEN 2 del SV-328, es el SCREEN 3 del MSX, es decir, el modo de baja resolución.

Las normas para su adaptación son las mismas que para el SCREEN 2 del MSX.

El truco siguiente es divertido y original. Podemos usarlo en la presentación de nuestros programas para visualizar las instrucciones. Al principio el programa nos pide un texto que posteriormente se imprimirá en la pantalla simulando una impresora. En el texto podemos incluir cualquier carácter, ya sea letra, número ó gráfico. Los caracteres «'» y «/» no se imprimirán, ya que son caracteres de control.

El siguiente «'» provocará un retorno a carro (continuando la impresión en la línea siguiente), por ejemplo el texto:

ESTO ES UNA DEMOSTRACION' DE LA RUTINA, se imprimirá como:

ESTO ES UNA DEMOSTRACION

DE LA RUTINA

El carácter «/» provoca un espacio en blanco, así el texto:

////////// HOLA ////// OTRA DEMO

se imprimirá como:

(10 espacios) HOLA (6 espacios) OTRA DEMO

Para incorporar la rutina a nuestros programas sustituiremos el INPUT de la línea 20 por:

A$ = «texto»

```
10   LOCATE 0,0,1:CLS
20 INPUT "INTODUZCA EL TEXTO ";A$
30 CLS
40 FOR C= 1 TO LEN(A$)+1
50 FOR A=1 TO 5:FOR B%=1 TO1:NEXT B%:OUT 170,255:OUT 170,0:NEXT A
60 IF MID$(A$,C,1)="'"THEN PRINT CHR$(13):NEXT C
61 IF MID$(A$,C,1)="\"THEN PRINT " ";:NEXT C
70 PRINT MID$(A$,C,1);:NEXT C
```

Invertir las letras

Les mando un pequeño programa que realiza la curiosa operación de poner las letras «patas arriba», dicho más técnicamente, invierte las letras en la pantalla.

Para ello me he servido de algunas direcciones de la VRAM y los conocidos VPOKEs y VPEEKs, sin embargo su acción es exclusiva de la pantalla modo texto 32 × 24 (SCREEN 1) y para volver a la normalidad basta con utilizar la antes mencionada instrucción SCREEN, con lo que la VRAM se carga de sus verdaderos valores. Sin embargo si lo que desea es una «escritura en chino», no hay más que omitir las líneas 90 y 100. El resultado es sorprendente, de efecto inmediato y que incluso con el programa borrado o fuera de él su acción permanece.

```
10 'INVERTIR CARACTERES DE PANTALLA
20 'por JOSE LUIS FERNANDEZ BARROS DE SAHAGUN (LEON)
30 'para TRUCOS MSX 18-9-85
40 SCREEN 1
50 FOR N=264 TO 2032 STEP 8
60 PRINT CHR$(N/8)
70 FOR D=7 TO 0 STEP-1
80 X(7-D)=VPEEK(N+D)
90 NEXT D
100 FOR D= 0 TO 7
110 VPOKE N+D,X(D)
120 NEXT D,N
```

José Luis Fernández Barros
Sahagún

Restaurar las teclas de función

En el complejo programa residente en ROM, nos encontramos con una rutina cuya misión es devolver la definición original a las teclas de función; dicha rutina se encuentra situada en la dirección 3Eh. Para utilizar dicha rutina basta con hacer una llamada a la dirección correspondiente mediante la función USR.


New Media Systems
PHILIPS MSX
Software MSX
Gestión y Productividad
Disfrute de las tareas de cada día, con un ordenador PHILIPS MSX y un programa de productividad
Un ordenador PHILIPS MSX hará que muchas de las gestiones cotidianas sean más llevaderas y más productivas. Como la preparación de la correspondencia, el análisis de informes, el mantenimiento de los ficheros de clientes y proveedores, la confección de presupuestos, la emisión de facturas y recibos, y en suma, la administración de pequeñas industrias y negocios.
El amplio surtido de programas MSX orientados hacia la gestión, son las soluciones que ahorran tiempo, esfuerzo y evitan errores en estos trabajos y muchos similares. La versión única de PHILIPS para el sistema operativo en disquette MSX-DOS, con su "auxiliar de usuario" hace más cómodo su trabajo ofreciendo la "selección por menú" de las operaciones a realizar y "páginas informativas" de ayuda al usuario.
Además de esta enorme variedad de programas de productividad, PHILIPS tiene mucho más que ofertar en el mundo de la informática: potentes lenguajes de programación como el Pascal, o como el inigualable y universalmente conocido MSX-LOGO, además de sus programas educativo-recreativos.
Los programas PHILIPS están respaldados por su gama de periféricos, incluyendo impresoras de calidad para correspondencia, unidades de disco de alta velocidad de transferencia y gran capacidad de almacenamiento, monitores monócromo y de color, cartuchos de interface serie, y muchos más.
LA NUEVA IMPRESORA DE MATRIZ PARA CORRESPONDENCIA PHILIPS VW0030 ES TOTALMENTE VERSATIL CON AMPLIA SELECCION DE TAMAÑOS, ESTILOS Y REALCES EN LA ESCRITURA, EN HOJAS SUELTAS O PAPEL CONTINUO.
LA UNIDAD DE DISCO DE ACCESO DIRECTO VY0010/11 CON DISKETTES DE 3.5" UNE A SU GRAN CAPACIDAD DE ALMACENAMIENTO UNA VELOCIDAD DE TRANSFERENCIA DE 250 KBITS POR SEGUNDO.
PHILIPS ofrece lo que el poderoso mundo del standard MSX merece: lo mejor. Y esto se concreta en sus equipos, en sus programas, y en el valor del dinero desembolsado. PHILIPS integra.
NMS
PHILIPS
Servicio de Información al simpatizante y usuario
Tel. (91) 413 21 62



DENSIDADES, BITS Y BITES

Las palabras «pulgadas, caras, pistas, sectores,...» nos son ya familiares con los ya antiguos diskettes de 5,25 pulgadas, pero cual es ahora nuestra sorpresa al descubrir en otro tipo de diskettes densidades diferentes a las simples o dobles. Este es el tema de la pregunta: ¿Qué es exactamente la densidad de un disco y cómo se relaciona con las caras, pulgadas, etc?

Xavier Casajuana Mogas
Barcelona

Su pregunta es muy interesante y demasiado compleja para poder responder en esta sección.

Es cierto que los fabricantes proponen su diskette como la opción ideal, ya sea de 3,5 ó de 5,25 pulgadas y ante la diversidad de aparatos se encuentre un poco desconcertado. De cualquier modo, para explicar lo que es la densidad de un disco tendremos que entrar en lo que son sectores, pistas y unos cuantos conceptos difíciles de tratar en tan poco espacio.

Tendremos en cuenta su posdata y procuraremos confeccionar un completo artículo sobre las unidades de discos que hay en el mercado.

PROBLEMAS CON ALGUNOS PROGRAMAS

Mi ordenador, por suerte o por desgracia, es un Spectravideo 328 MK-II. La mayoría de las cosas son compatibles con el MSX, pero hay otras que no lo son y ahí me atasco, aunque hay cosas que puedo traducir. Me gustaría que me ayudaran a subsanar estas diferencias.

Santiago Seguro Merino
Granada

Efectivamente, debido a que muchos usuarios actualmente poséen el SV-328, hemos confeccionado una pequeña tabla de conversión entre los dos sistemas (ver Sección de Trucos e ideas de este mes). Como podrás comprobar la principal diferencia radica en los modos de pantalla de ambos ordenadores. Esto habrá que alterarlo debidamente para que cualquier programa de MSX funcione correctamente en el SV-328.

PROBLEMAS CON UN LISTADO

En el número de septiembre publicábais una carta en la que se dice que los programas son probados varias veces sobre lo cual albergo serias dudas, ya que en el programa «LOS VISITANTES DE MADRID» se cometen errores de bulto, como igualar variables numéricas con alfanuméricas en la línea 450 (450 S=INPUT$ (1)).

Juan Pastor
Sevilla

Volvemos a insistir en que los programas publicados son probados y revisados sus listados varias veces. Respecto a la línea que nos menciona podemos asegurarle que no existe ningún tipo de error en ella, puesto que INPUT$ (1) no se refiere a una variable alfanumérica, sino a una instrucción BASIC cuya función es leer un número del teclado y colocarlo en la variable S.

DIRECTOR:
Juan Arencibia
COORDINADOR EDITORIAL:
J. Ignacio Rey
REDACCION:
Fernando García, Santiago Gala, Ricardo García, Teresa Aranda,
DISEÑO:
Ricardo Segura y Benito Gil.
Editada por
PUBLINFORMATICA S.A.
PRESIDENTE:
Fernando Bolín.
DIRECTOR EDITORIAL:
Norberto Gallego.
Administración:
PUBLINFORMATICA
GERENTE DE CIRCULACION Y VENTAS:
Luis Carrero.
PRODUCCION:
Miguel Onieva.
DIRECTOR DE MARKETING:
Antonio Gonzalez.
SERVICIO AL CLIENTE:
Julia Gonzalez.
Tel. 733 79 69
ADMINISTRACION:
Miguel Atance,
JEFE DE PUBLICIDAD:
María José Martín.
DIRECCION Y REDACCION:
C/ Bravo Murillo, 377 - 5° A
Tel. 733 74 13
28020 MADRID
PUBLICIDAD Y ADMINISTRACCION:
C/ Bravo Murillo 377 - 3° E
Tel. 733 96 62-96
Publicidad en Madrid:
Fernando Hernando
Publicidad en Barcelona:
Olga Martorell
C/ Pelayo, 12
Tel. (93) 301 47 00 Ext. 27-28.
08001 BARCELONA
Deposito Legal: M. 16.755-1985
Impreso en Héroes S.A.
C/ Torrelara, 8. 28016-MADRID.
Distribuye:
S.G.E.L. Avda. Valdelaparra s/n.
Alcobendas (Madrid).
DISTRIBUIDORES:
PORTUGAL: VASP Sociedad de Transportes e Distribuidores. LISBOA
CHILE: CORPRODE SANTIAGO
ARGENTINA: DISTRIBUIDORA INTERCONTINENTAL BUENOS AIRES

SUSCRIPCIONES:
Rogamos dirija toda la correspondencia relacionada con suscripciones a:
MSX
EDISA: Tel. 415 97 12
C/López de Hoyos, 141-5.°
28002 MADRID
(Para todos los pagos reseñar solamente MSX)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
MSX
C/Bravo Murillo, 377-5.° A
Tel. 733 74 13 28020 MADRID

Si deseas colaborar en MSX remite tus artículos o programas a Bravo Murillo 377, 5.° A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.

New Media Systems

Aprender puede ser divertido, con un ordenador PHILIPS MSX y el lenguaje de programación LOGO.

Porque LOGO ha sido especialmente desarrollado para permitir a los jóvenes usuarios, iniciarse rápidamente en informática. Es increiblemente fácil de usar, con sencillos comandos en el idioma "nativo" del usuario.

Además el MSX-LOGO de PHILIPS le da acción con color y sonido; lo que ayuda a generar y mantener su atracción en el aprendizaje. Por eso el PHILIPS MSX-LOGO es el favorito en todos los colegios y escuelas del mundo. Y aunque LOGO es recreativo, también constituye el fundamento sobre el cual el usuario adquirirá mayor destreza en la solución de problemas, y es en definitiva el sólido y excelente cimiento para estudios más avanzados.

Como el propio Dr. Seymour Papert, inventor del LOGO declaró: "MSX y LOGO forman el matrimonio más ideal que podamos imaginar. Estando ambos orientados a gráficos, el MSX apoya todo lo que el LOGO ha de ofrecer en cuanto a música y animación de figuras".

LA LECTO-GRABADORA DE DATOS EN CINTA CASSETTE CONSTITUYE EL EQUIPO IDEAL DE BAJO COSTE PARA EL ALMACENAMIENTO DE INFORMACION -DATOS Y PROGRAMA- EN TODOS LOS ORDENADORES MSX.

EL ORDENADOR PHILIPS POSEE FACILIDADES EXTRAORDINARIAS PARA LAS IMAGENES EN COLOR, Y FORMA LA BASE PARA UN SISTEMA PERFECTO Y AMPLIABLE QUE CUMPLE LA NORMA STANDARD MUNDIAL MSX.

Los ordenadores PHILIPS MSX disponen de una total capacidad gráfica en color y generación de música, y se integran de forma ideal con las facilidades educativas del LOGO.

El MSX-LOGO de PHILIPS es el único que puede presentar hasta 30 tortugas, y cada una adoptando una figura entre 60 definibles por el usuario mediante el editor incorporado; admite hasta 16 colores, puede emitir por tres canales musicales y otro más para efectos sonoros, el movimiento de figuras es autónomo, detecta choques de 'tortugas' y otros eventos, realiza un completo tratamiento de LISTAS y propiedades.

PHILIPS ofrece lo que el poderoso mundo del Stándard MSX merece: lo mejor. Y esto se concreta en sus equipos, en sus programas, y en el valor del dinero desembolsado. PHILIPS integra.



PHILIPS

PHILIPS

Servicio de Información al simpatizante y usuario
Tel. (91) 413 21 62



DISFRUTELO EN FAMILIA

El ordenador personal para toda la familia. Con 80 K para estar a la altura de todos los gustos y necesidades. Jugar, archivar, aprender, programar: y con capacidad para crecer con la aplicación de periféricos.

CARACTERISTICAS MAS IMPORTANTES DEL V-20

- Sistema standard MSX que hace compatibles hardware y software de todos los productos que tienen este sistema en el mercado.
- Pueden acoplar los siguientes periféricos de CANON:
  - — Impresoras.
  - — Floppy de 720 K, que incluye diskette MSX-DOS para aprovechar toda la capacidad del ordenador y además incluimos un segundo diskette con tres programas de aplicaciones profesionales.
  - — Mouse para hacer todo tipo de gráficos a color.
  - — Joy sticks.
  - — Caja de 5 diskettes vírgenes.
  - — Variedad de programas en cinta con juegos.
- Y además dos manuales en castellano: guía del usuario y completo manual de BASIC.